Kwisatz

Enabling Activity Context

Anonymous Author(s)

ABSTRACT

Our ability to find digital data is reaching a tipping point: brute force search techniques are inefficient and searching multiple storage locations to find related objects is challenging. Prior research found using contextual clues facilitates finding specific digital objects. Despite modern systems collecting vast amounts of contextual information, our systems do not provide an efficient mechanism for using that information to facilitate more efficient *finding* of digital objects.

Kwisatz is our system for collecting, storing, and disseminating contextual information we call *activity context* to facilitate finding groups of related digital objects regardless of where those objects are stored. We find Kwisatz is a viable way to provide *activity context* and enabling its use by other services and applications.

ACM Reference Format:

Anonymous Author(s). 2022. *Kwisatz*: Enabling Activity Context. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

TM ►*TBD*◄

2 BACKGROUND

TM ►*TBD*◄

3 ARCHITECTURE

Kwisatz is logically composed of three major components, each of which is an essential portion of providing the end-to-end systems level support for capturing, storing, and utilizing *activity data*. Each of these components consists of various smaller components assembled together to provide the necessary services.

In this section, I lay out the basic architecture of these components. In subsequent sections, I will drill down into this architecture and identify key aspects of the system design. In constructing this architecture, I have attempted to broadly address what I consider to be the key aspects of these components, including defining terminology and identifying potential use cases that may be relevant. Frequently, I will suggest potential implementations that would fit within this architecture: there is no expectation that I will implement even a majority of these potential implementations. Rather, the goal of using broad considerations is to assist in ensuring the system architecture is reasonably flexible. Ultimately, my goal is to demonstrate the architecture is itself viable as a system service.

- **Ingestion** — the activity data that are presented by various services within the system needs to support a rich and robust model in which captured data may be converted into a common form that permits utilization. §3.1

Conference'17, July 2017, Washington, DC, USA

2022. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

https://doi.org/10.1145/nnnnnnn.nnnnnnn

Figure 1: *Kwisatz* Architecture Diagram TODO

- **Storage** — raw activity data, along with extrapolated activity meta-data, need to be stored in a format that is scalable and efficient as well as supporting a robust utilization model. §3.2
- **Utilization** — to realize the benefits of activity data, the system must have a useful model for using the activity data. §3.3

3.1 Ingestion

Activity data can arise from a variety of sources. For example, because my primary focus is on utilizing activity data to better inform logical data organization, I view data storage as being a key source of such information. Indeed, much of the prior work in this area has focused on utilizing information about data, including:

- **File Names** — the *file name* is a time honored way to embed information about the file itself. For example, in *Burrito* [106] the observation is that we capture parameter information within the file name. This is because it is the *only* way to safely capture this information in a way that is broadly viable across file systems.
- **Extended Attributes** — the *extended attribute* is a mechanism that provided a way for applications to create additional meta-data using an attribute/key model [197]. File systems that support extended attributes maintain them as meta-data of the file itself. Unfortunately, extended attributes suffer from two limitations: (1) file systems that do support them provide no mechanism for associating files based upon the extended attributes; (2) there is no uniform support or implementation of extended attributes.
- **File Meta-data** — most file systems support at least a minimal subset of meta-data elements, specifically timestamps and size. There is a lack of uniformity of other attributes: POSIX file systems typically support "mode" bits that represent access permissions as well as potentially other behaviors, as well as an access time, modification time, and change time. Creation timestamps are often maintained by file systems as well. A number of UNIX-like file systems maintain a creation timestamp. Windows includes the *creation* time of the file and that has been adopted by a number of UNIX-derived file systems, including ext4, jfs, and btrfs. Recent changes in Linux include a new system call, **statx** that permits applications to retrieve this information programmatically.
- **Directories** — the traditional hierarchical file system provides a mechanism for composing groups of files into a *directory*, which is a set of files. Some file systems restrict a file to being a member of a single directory, while others allow a

file to be a member of multiple directories. The Mutics file system included the ability to create links to files. Modern file systems may implement links as either direct references from the directory to the file ("hard links") or indirect references from an entry in the directory to the file ("soft links.") They have somewhat different semantics.

Views — in semantic file systems [95] there is less emphasis on reference counted relationships (e.g., hard links) or even persistence and more emphasis on creating logical groups ("virtual directories") of files based upon some criteria.

Kwisatz does not seek to *replace* any of these existing file storage mechanisms. Instead, it focuses on providing rich support for meta-data *about* digital objects, which includes files but should not be limited strictly to files.

Meta-data is not strictly limited to the data that is available from the file system itself. Further, meta-data may be about digital objects that *existed* at some point in time but no longer exist: this is a reality of separating the storage from the meta-data service. Instead of focusing on maintaining a strongly referenced model, I instead adopt the model of the Internet, which means that meta-data may reference digital objects that no longer exist. Applications that use *Kwisatz* should be aware that the underlying data could cease to exist and act accordingly.

Other potential sources of meta-data are quite broad and include:

Semantic transducers — the term *transducer* was first introduced as part of semantic file systems [95]. The basic idea was that active components would *extract* semantic information from the contents of a file and then use it for indexing. Indeed, modern indexers work on this basic principle, without making any changes to the file system.

Content classifiers — one common use for machine learning is to identify the content of specific files, such as images or videos, to determine if the given file contains specific content: a cat, for example. Such classifiers can be more targeted, such as finding pictures of a specific person, or containing a *particular* cat. This information can then be used to cluster files together, such as the "video reels" that some service providers now give us on our personal devices.

Hashing — a *hash* can be computed on a given file to determine if the contents of said file has changed. For example, this can be used by "cloud storage" providers to determine if a given file has already been uploaded. Hashing can also be used to determine when files have changed.

Metrics — one common approach to information is to establish the logical proximity of the files, whether based upon the *content* of the file, or the *meta-data* of the file. For example, plagiarism tools like MOSS look for structural similarity (versus simpler textual similarity) by comparing the abstract syntax trees of code. This generates a measure of similarity. The *value* of the metric is not material to this project, but the *use* of metrics is because it allows us to create a logical distance between the objects. This can, in turn, be used to *cluster* objects that are "close to" each other.

Environment — our devices maintain multiple sources of environmental information. For example, location information (*GIS*) identifies where a device is located at a given time. Increasingly, our devices track other aspects of the environment, including the ambient temperature, our vital measurements such as heart rate and blood pressure, and even more detailed health information including data from pace makers, insulin pumps, and menstrual cycle trackers. The data from these is likely useful in creating associations between extrinsic events and human storage usage.

Social — our devices routinely track our social activity: with whom we are interacting via text, chat, e-mail, and video call, for example. Frequently, as part of this we share information — information that is subsequently stored, modified, and re-shared onwards. This type of activity data can be used to help us identify where information came from, what other digital objects were accessed as a result, and establishing data relationships based upon usage patterns. Web sites visited, Reddit posts liked, Discord messages exchanged, music listened to, purchases made, and even games played can all be used to construct associative relationships that make sense to human users.

In general, activity data can be *intrinsic* to the digital object, such as information based upon its semantic content, it's length, and contents as well as *extrinsic* to the digital object, such as what applications were used to access it, where things were done, with whom, etc. The *Kwisatz* architecture is influenced by my desire to ensure support for a broad range of activity data sources.

In §4 I delve into greater detail about handling activity data.

3.2 Storage

The choice of storage models, while important, is unlikely to give rise to significant research questions at this juncture: as we begin to understand the nature of the activity data we anticipate collecting, it is distinctly possible that new challenges will emerge. However, we have an extensive body of knowledge on how to handle high data rates ("drinking from the fire hose") as well as scaling approaches for managing potentially large bodies of data.

Thus, while the architecture is fairly neutral with respect to the details here, I anticipate the initial implementation of this will utilize "reasonable" limitations on activity data sources (e.g., curation to avoid excessive data loads).

An important aspect of the architecture is to propose a model for the data format that I propose using. This data format must be able to:

- Identify the *source* of the activity data. This permits interpretation of the captured data by any transducer familiar with the data generated by the given source.
- Specify the *version* of the activity data. My own review of numerous data sources suggests that it is common for many of them to change the format of the data over time; typically this *extends* the data format (common for systems-related activity data sources, for example) but sometimes it involves significant restructuring of the data that is available (common for web-based activity data sources.) By including a version, a transducer can determine if it understand the format of this data and permits evolution.
- Provide an ordering of the activity data. Typically this would be a "timestamp," though there is no reason this

needs to be a timestamp relative to any other data source. Further, when there are multiple providers of information, the interpretation of this Lamport clock is ultimately determined by the transducer. This *allows* both system relative and universal clocks but does not dictate their presence nor disallow clocks that are shared between activity data providers.

- A list of *attributes*. These are in the format of "extended attributes," with both an identifier as well as a value. This permits attributes that can be the same across sources, as well as allow the same "attribute" to have different meanings for different sources. This is neither required nor prohibited within the *Kwisatz* architecture. This ability provides very broad support for activity data, as well as *post hoc* supplementation by transducers.
- The *raw data* originally captured by the activity data provider. This allows the capture of information without requiring interpretation at the time it is captured. In cases where there is no additional raw data, this can remain empty. Note that this is *not* anticipated as being an area in which to store the digital object's data.

As a concrete example, I have implemented an activity data provider that scans and captures the change data from the NTFS USN Journal on a Windows 11 computer. The raw data is captured, and then certain elements of the data can be augmented. For instance, the raw data provides a *file id* that is used to obtain the name of the file. Similarly, the raw data also provides a *directory id* that is used to obtain the path of the containing directory. This is relative to the NTFS volume (which does *not* include a drive letter.) Thus, the transducer for this can utilize the *volume id* to map to the current drive letter, making this name available for ordinary Windows applications (which tend to use drive letters, even though they aren't visible to the NTFS file system controlling a given volume.)

Subsequent to this, I envision a separate transducer that can be used to compute the hash value of the file's contents. That hash value could then be incorporated into the attributes list of the corresponding change journal record (assuming the file has not changed by the time the hash value has been computed, of course.)

This data format can then be easily captured as a JSON expression, which can then be used to store the relevant data in a database (e.g., DynamoDB or MongoDB, for example.) This decouples the specific details of how the data is stored from the format required of the data gathering elements.

I discuss this in greater detail in §5.

3.3 Utilization

It is essential to have a model for utilizing activity data in order to realize its potential. The *Kwisatz* architecture is generalized with an eye towards permitting a range of potential use cases.

Underlying the structure of the data are elements that I anticipate will be used to establish relationships. For example, the *metric* concept described earlier (§3.1) naturally fits with data clustering mechanisms. This is consistent with a *graph* representation in which the edges correspond to some relationship and the weight of the edge corresponds to a metric. Similarly, it can be useful to add labels to the data, in order to understand specific characteristics of that information. The emphasis on using graph modeled storage (§3.2) is also consistent with this. Thus, the logical way to utilize this information is to focus on exploiting the inherent graph structure of the data.

Given this model of exploring graph data, it makes sense to then consider using an existing graph query language — reserving the right to limit its use to some subset as part of the exploratory work I am doing. One motivation for this would be that such languages *already* work with commonly used graph databases and thus leverage prior work in using graph structured information effectively.

I discuss this in greater detail in §6.

4 INGESTION

There is no single mechanism for retrieving activity data because each source for such data provides it in multiple different ways. Thus, for the purposes of our analysis we found it useful to categorize data sources based upon how we retrieved the information. Thus, one of the jobs of an ingestion agent is to retrieve the data and then put it into our common format, which will permit us to use that information for subsequent processing and analysis.

4.1 Models

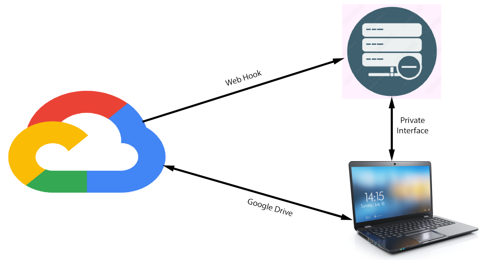
In our work we have found multiple ways that activity data can be retrieved:

Systems API — one common approach for retrieving activity data is via an operating system defined application programming interface (API). For example, the eBPF mechanism commonly found in several operating systems represents a general mechanism for retrieving activity data from a variety of sources. To that extent it provides a general extensible framework for activity data, with the format of the data being returned dependent upon its source. Similarly Microsoft Windows has an operating system mechanism known as *Event Tracing for Windows* that is used extensively by Microsoft to provide detailed activity data for essentially all portions of their operating system. A more focused interface on Windows is the USN Journal, which is supported by the NTFS file system. Each of these provides a specific interface for retrieving data that is directly visible to the local system.

Web APIs — another common approach is to provide web interfaces to service providers that can be used to retrieve information, including activity data. Such web APIs may be directly accessible on the local system or may involve contacting a remote service provider. In each of these cases an agent can capture this data and store the relevant activity data in our common database. The specific details of the web API vary by provider. For example, the Dropbox service on a local system supports querying for changes via a web API against the local system. The Github API includes the ability to query for events on a polling basis.

Web hooks — some web services do not provide a direct event query interface and instead require creation of an intermediate agent via a registered web hook. In such a case, the web service provider is given the address of a distinct publicly available web service which is invoked

Figure 2: Google Drive API Web Hook



periodically with event information. Because the web hook must be publicly available an agent collecting data from such a service requires both a local agent on the machine where the activity data is being recorded as well as a public agent on an internet facing public machine.

Manual Scanning — some services do not provide any simple mechanism for detecting activity data directly. In such a case it is possible to construct an activity agent that manually scans for changes. One example if this would be the *rclone* facility which uses brute force scanning of a storage location to determine "what has changed" and then uses that information to synchronize copies of that data on a secondary copy. While simple, this approach is well-known and can be used when building a activity data gathering agent.

These various techniques are all *possible* but have potentially radically different performance characteristics that may limit their utility. Thus, we wanted to construct at least one example of each in order to better understand the viability of using activity data collection agents for each of these classes of service.

In some cases we found it was possible to create indirect monitoring mechanisms. For example, one service we commonly use — *Overleaf* — does not have a web monitoring API. However, it *does* integrate with Github repositories and thus we can simulate tracking Overleaf change data by using the Github APIs.

Another approach, which we did not explore for this work, would be to use API aggregation services such as Zappier. Such aggregation sites combine web APIs and may be a fruitful way to extract additional activity data from a broad range of sources.

We chose a single example of each of these types of activity data retrieval. This allows us to explore the space broadly and gain further insight into the viability of using such sources.

- For our systems API example we chose to use the Update Sequence Number (USN) Journal support provided by the NTFS File system in Windows [120]. This system provides a curated view of file system activity on drives where the USN journal has been enabled. By default the USN journal is enabled on the system drive and may be optionally enabled on other drives. This is a publicly documented API, has good performance, and provides a curated level of information that provides benefit without being "too broad."
- For our Web API example we use the Dropbox cloud storage system. This relies upon a local service on a specific computer. This service can be used to monitor activity on the Dropbox system. We used a standard Dropbox installation on Windows, Version 154.4.5353. TM ►*Add citation for Dropbox?*◄
- For our Web Hook example we use the Google Drive API. To do this we constructed a public facing web service that we could register with Google to receive changes (the "web hook"). We also constructed a local service that could run on a desktop computer to obtain the necessary credentials and provide them to Google as well as gather up data from our public facing web service. We show a simple diagram of this arrangement in Figure 2. Our implementation was simple and relied upon pre-installed certificates between client and webhook to provide cross-system authentication.
- For our manual scanning example, we chose to use TM ►*TBD*◄.

While building the various activity data collection agents we made a number of observations around potential challenges to constructing such agents:

- It is broadly useful to have *curation* over the data source. One of the challenges of reading from a "fire hose" is that the "signal-to-noise" ratio can be low. Instead, the emphasis should be on capturing discrete events that are useful.
- To use labels to identify the specifics of particular relationships and properties to capture information that can be used later requires a level of understanding of what the data source provides and how that compares with other data sources. For example, timestamps might be collected using a different interpretation of time zero so to use this information requires some means of comparing such cross-domain values. Similarly, in a system where multiple objects are grouped together via the naming system (e.g., files within a directory) it is useful to capture that relationship when the activity agent is recording information.
- TM ►*TBD*◄

4.2 Evaluation

Our evaluation of these activity data collection agents involves considering several key factors:

- The time and complexity of the code needed to implement the agent.
- The amount of data the agent generates per unit time.
- The performance impact of the agents on the system.
- TM ►*TBD*◄

TM ►*So, the point here is to provide a straight-forward framework for evaluating the overhead associated with these activities as part of the evaluation.*◄

5 STORAGE

A "labeled property graph" includes:

node — in traditional mathematical graph descriptions this is the *vertex*. For my work this corresponds to a digital object, such as a file in a file system, a value in a key/value store, etc. Nodes within my/ system have *structure* because it must codify "where" the actual digital object is located in addition to the other information needed. For practical purposes I think of the "location" as being a uniform resource identifier (URI.) The benefit of this paradigm is that it relies upon an existing standard for identifying objects, which is sufficient for my work.

relationship — in traditional mathematical graph descriptions this is the *edge* that connects a pair of vertices.

identifier — this identifies the graph element (node or relationship).

label — this is a descriptive property related to the node or relationship. For example, a label might identify that a node is a *file* versus a *value* in a key/value store. Similarly, a label associated with a relationship would establish *what* type of relationship this represents, such as a *derived from* or a *contained by* relationship.

property — this is essentially a key/value pair, where the key represents the property and the value represents the data associated with that property. My goal is to permit some structure to a property, so that it may have a version associated with it, permitting more flexible upgrades to the format of the data.

I choose this model because it fits the data we collect and permits exploration of augmenting the model with distinct agents capable of adding specific property information as well as relationships.

Thus, *nodes* in this model represent data objects and activity events. This can be used to create different types of relationships. For example, *causal* relationships would show a flow from one data object to a second data object based upon an activity event that "explains" that transformation. This would be a way to capture *provenance* information. However, the model is not limited to this type of relationship. For example, data objects created when the same web page was being accessed could be identified by using an activity event node representing the web page access. This sort of contextual relationship is useful for finding related data objects that are not causally linked. This might occur, for example, when writing code. A developer could be shown other files they created when looking at the same page, allowing them to find that prior work.

Because our goal is to find assocations via explicit and inferred contextual information, we describe activity *events* and *contexts*. An activity *event* is a node that an activity provider inserts to represent some specific information about *what* happened. An activity *context* is created on demand and associates a set of activity events with another object (e.g., a data object). For a data object, this creates a relationship between the object and potentially related activity events.

Thus, the expectation is that we can retrieve the most recent instance of an activity event of interest (where I leave defining *of interest* to be determined.) For a simple demonstration, it likely makes sense to simply have a list of the "most recently added" instance for each activity provider and simply form the activity context by associating it. As the number of activity providers increases, further work will likely be needed to enable scaling.

Thus, an *activity context* can be associated with other objects. This will permit us to subsequently examine the context of operations that were ongoing at the time of interesting events.

This leads to a number of interesting questions to explore:

- Does it make sense to maintain temporal relationships in this fashion; in other words, does the newest activity context point back to the previous activity context, or do the activity events point back to the older activity event?
- Should an activity context be something added implicitly (e.g., each time one creates a *data* node, should the system automatically associate an activity context with it,) or explicitly (e.g., only when performed by an external agent.)
- How do we garbage collect this information? That's not likely an important question to answer for prototypes, but it will become an important question as systems such as this emerge.

5.1 Use Cases

While we expect numerous potential uses of activity context to be found, we have identified three interesting cases that we have used to drive our own exploration of activity context.

Search — search services are ubiquitous in storage solutions. Prior work has clearly identified that activity context information can be used to improve search outcomes [286]. Further, the ability to search across distinct storage domains provides benefits to users over the iterative search mechanisms they typically use.

Data Browser — our group did preliminary experimentation with an alternative file browser that worked using associations, rather than limiting themselves to hierarchical structure. This is also reminiscent of work done in the Human-Computer Interface (HCI) community regarding collaborative non-hierarchical data organization tools [63]. Our richer contextual environment is one that supports further exploration of alternative data visualization techniques.

TBD — TM ►*Need to find some other good use case here.*◄

5.2 Evaluation

6 UTILIZATION

6.1 Use Cases

6.2 Evaluation

7 CONCLUSION

TM ►*The idea that using behavioral advertising information could be beneficial to file search would potentially be of interest to aggregation companies because it would encourage use of these tools willingly: a "free" service subsidized by advertising.*◄

REFERENCES

[1] Atul Adya, William J. Bolosky, Miguel Castro, Gerald Cermak, Ronnie Chaiken, John R. Douceur, Jon Howell, Jacob R. Lorch, Marvin Theimer, and Roger P. Wattenhofer. 2002. Farsite: federated, available, and reliable storage for an incompletely trusted environment. In *Proceedings of the 5th symposium on Operating systems design and implementation (Copyright restrictions prevent ACM from being able to make the PDFs for this conference available for downloading)*, Vol. 36. 1–14.

[2] Atul Adya, William J. Bolosky, Miguel Castro, Gerald Cermak, Ronnie Chaiken, John R. Douceur, Jon Howell, Jacob R. Lorch, Marvin Theimer, and Roger P. Wattenhofer. 2003. Farsite: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment. *SIGOPS Oper. Syst. Rev.* 36, SI (Dec. 2003), 1–14. https://doi.org/10.1145/844128.844130

[3] Bryan Agee, Charlie Daly, and Josh Brown. 2016. Ubuntu A tag-based filesystem for ubuntu. (2016), 1–3.

[4] Nehad Albadri, Richard Watson, and Stijn Dekeyser. 2016. TreeTags: Bringing Tags to the Hierarchical File System. In *Proceedings of the Australasian Computer Science Week Multiconference* (Canberra, Australia) *(ACSW '16)*. Association for Computing Machinery, New York, NY, USA, Article 21, 10 pages. https://doi.org/10.1145/2843043.2843868

[5] Amazon Web Services. 2021. Editing Object Metadata. Online. Accessed 2021-02-01. https://docs.aws.amazon.com/AmazonS3/latest/user-guide/add-object-metadata.html.

[6] Inc. Amazon Web Services. 2018. Object Tagging. https://docs.aws.amazon.com/AmazonS3/latest/dev/object-tagging.html (accessed January 8, 2019).

[7] Amazon Web Services, Inc. 2021. Amazon Simple Storage Service Developer Guide. API Version 2006-03-01. Online. Accessed 2021-02-01. https://docs.aws.amazon.com/AmazonS3/latest/dev/s3-dg.pdf.

[8] Sasha Ames, Nikhil Bobb, Kevin M Greenan, Owen S Hofmann, Mark W Storer, Carlos Maltzahn, Ethan L Miller, and Scott A Brandt. 2006. LiFS: An attribute-rich file system for storage class memories. In *Proceedings of the 23rd IEEE/14th NASA Goddard Conference on Mass Storage Systems and Technologies*. IEEE.

[9] Sasha Ames, Maya Gokhale, and Carlos Maltzahn. 2013. QMDS: a file system metadata management service supporting a graph data model-based query language. *International Journal of Parallel, Emergent and Distributed Systems* 28, 2 (2013), 159–183.

[10] Pierre Andrews, Ilya Zaihrayeu, and Juan Pane. 2012. A classification of semantic annotation systems. *Semantic Web* 3, 3 (2012), 223–248. https://doi.org/10.3233/SW-2011-0056

[11] Renzo Angles, Marcelo Arenas, Pablo Barcelo, Peter Boncz, George Fletcher, Claudio Gutierrez, Tobias Lindaaker, Marcus Paradies, Stefan Plantikow, Juan Sequeda, et al. 2018. G-CORE: A core for future graph query languages. In *Proceedings of the 2018 International Conference on Management of Data*. ACM, 1421–1432.

[299]]apple:spotlight-extensions Apple. [n. d.]. Search and Spotlight. https://tinyurl.com/yasf7kf2 (accessed January 6, 2019).

[13] Apple, Inc. 2016. Search Drives User Engagement.

[14] Raja Appuswamy. 2014. Building a File-Based Storage Stack: Modularity and Flexibility in Loris. (2014).

[15] R. Appuswamy, D.C. van Moolenbroek, and A.S. Tanenbaum. 2010. Loris - A Redundant Array of Independent Physical Layers. In *Proceedings of the sixteenth annual conference of the Advanced School for Computing and Imaging (ASCI'10)*. Advanced School for Computing and Imaging (ASCI).

[16] Marcelo Arenas, Bernardo Cuenca Grau, Evgeny Kharlamov, Šarūnas Marciuška, and Dmitriy Zheleznyakov. 2016. Faceted search over RDF-based knowledge graphs. *Web Semantics: Science, Services and Agents on the World Wide Web* 37 (2016), 55–74.

[17] Remzi Arpaci-Dusseau and Andrea Arpaci-Dusseau. 2021. *Naming and binding of objects*. Chapter 41. http://pages.cs.wisc.edu/~remzi/OSTEP/file-ffs.pdf

[18] Vaggelis Atlidakis, Jeremy Andrus, Roxana Geambasu, Dimitris Mitropoulos, and Jason Nieh. 2016. POSIX Abstractions in Modern Operating Systems: The Old, the New, and the Missing. In *Proceedings of the Eleventh European Conference on Computer Systems* (London, United Kingdom) *(EuroSys '16)*. Association for Computing Machinery, New York, NY, USA, Article 19, 17 pages. https://doi.org/10.1145/2901318.2901350

[299]]ashish Anonymized authors. [n. d.]. Anonymized title.

[20] Ajay Bakre, Dhruva Krishnamurthy, Kartheek Muthyala, Chhavi Sharma, and Rukma Talwadker. 2021. Federated namespace of heterogeneous storage system namespaces. Patent US10812313B2. Online. Accessed 2021-02-01. https://patents.google.com/patent/US10812313B2/en.

[21] Igor V Balabine, Ramiah Kandasamy, and John A Skier. 1999. File system interface to a database. US Patent 5,937,406.

[22] Igor V Balabine, Ramiah Kandasamy, and John A Skier. 2002. Database interface for database unaware applications. US Patent 6,442,548.

[23] G A III Barnard and Louis Fein. 1958. AN INFORMATION FILING AND RETRIEVAL SYSTEM FOR THE ENGINEERING AND MANAGEMENT RECORDS OF A LARGE-SCALE COMPUTER DEVELOPMENT PROJECT. *American Documentation (pre-1986)* 9, 3 (07 1958), 208. https://search.proquest.com/docview/195441816 Copyright - Copyright Wiley Periodicals Inc. Jul 1958; Last updated - 2010-08-27.

[24] Deborah Barreau and Bonnie A. Nardi. 1995. Finding and reminding: file organization from the desktop. *ACM Sigchi Bulletin* 27, 3 (1995), 39–43.

[25] Doug Beaver, Sanjeev Kumar, Harry C Li, Jason Sobel, Peter Vajgel, et al. 2010. Finding a Needle in Haystack: Facebook's Photo Storage.. In *OSDI*, Vol. 10. 1–8.

[299]]id3fs Ian Beckwith. [n. d.]. id3fs. ([n. d.]). https://erislabs.net/ianb/projects/id3fs/id3fs-index.html

[27] Introduction Motivation Benefits, Risks Current, File Systems, Data Anyone, Design Space, Integrated Approach, Augmented Approach, Description Canonical, and Architecture Comparison. 2016. Semantic File Systems. (2016), 1–17.

[28] Juan Benet. 2014. IPFS - Content Addressed, Versioned, P2P File System. *arXiv preprint arXiv:1407.3561* (2014).

[29] Ofer Bergman, Tamar Israeli, and Steve Whittaker. 2019. Factors hindering shared files retrieval. *Aslib Journal of Information Management* 72, 1 (2019), 130–147.

[30] Ofer Bergman, Tamar Israeli, and Steve Whittaker. 2019. Search is the future? The young search less for files. In *Proceedings of the Association for Information Science and Technology*, Vol. 56. 360–363.

[31] David Bermbach, Markus Klems, Stefan Tai, and Michael Menzel. 2011. MetaStorage: A Federated Cloud Storage System to Manage Consistency-Latency Tradeoffs. In *2011 IEEE 4th International Conference on Cloud Computing*. 452–459. https://doi.org/10.1109/CLOUD.2011.62

[32] T. Berners-Lee, R. Fielding, and L. Masinter. 1998. Uniform Resource Identifiers (URI): Generic Syntax. *RFC* 2396 (1998), 1–40.

[33] Alysson Neves Bessani, Ricardo Mendes, Tiago Oliveira, Nuno Ferreira Neves, Miguel Correia, Marcelo Pasin, and Paulo Verissimo. 2014. SCFS: A Shared Cloud-backed File System.. In *USENIX Annual Technical Conference*. Citeseer, 169–180.

[34] G. Beydoun. 2009. Formal concept analysis for an e-learning semantic web. *Expert Systems with Applications* 36, 8 (2009), 10952–10961. https://doi.org/10.1016/j.eswa.2009.02.023

[35] K. Birman and T. Joseph. 1987. Exploiting virtual synchrony in distributed systems. In *Proceedings of the eleventh ACM Symposium on Operating systems principles*, Vol. 21. 123–138.

[36] Andrew D. Birrell, Roy Levin, Michael D. Schroeder, and Roger M. Needham. 1982. Grapevine: an exercise in distributed computing. *Communications of The ACM* 25, 4 (1982), 260–274.

[37] Stephan Bloehdorn, Olaf Görlitz, Simon Schenk, and Max Völkel. 2006. TagFS - Tag Semantics for Hierarchical File Systems. In *Proceedings of the 6th International Conference on Knowledge Management (I-KNOW 06), Graz, Austria, September 6-8, 2006*.

[38] Stephan Bloehdorn, Olaf Görlitz, Simon Schenk, Max Völkel, et al. 2006. Tagfs-tag semantics for hierarchical file systems. In *Proceedings of the 6th International Conference on Knowledge Management (I-KNOW 06), Graz, Austria*, Vol. 8. 6–8.

[39] Richard Boardman, Robert Spence, and M Angela Sasse. 2003. Too many hierarchies? The daily struggle for control of the workspace. In *Proceedings of HCI international*, Vol. 1. 616–620.

[40] Cristiana Bolchini, Carlo A Curino, Elisa Quintarelli, Fabio A Schreiber, and Letizia Tanca. 2007. A data-oriented survey of context models. *ACM Sigmod Record* 36, 4 (2007), 19–26.

[41] Dhruba Borthakur. 2014. RocksDB: A persistent key-value store.

[42] Antal Bosch, Toine Bogers, and Maurice Kunder. 2016. Estimating search engine index size variability: a 9-year longitudinal study. *Scientometrics* 107, 2 (2016), 839–856.

[43] Cécile Bothorel, Juan David Cruz, Matteo Magnani, and Barbora Micenkova. 2015. Clustering attributed graphs: models, measures and methods. *Network Science* 3, 3 (2015), 408–444.

[44] Peter Braam. 2019. The Lustre storage architecture. *arXiv preprint arXiv:1903.01955* (2019).

[45] Will Brackenbury, Rui Liu, Mainack Mondal, Aaron J. Elmore, Blase Ur, Kyle Chard, and Michael J. Franklin. 2018. Draining the Data Swamp: A Similarity-Based Approach. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics* (Houston, TX, USA) *(HILDA'18)*. Association for Computing Machinery, New York, NY, USA, Article 13, 7 pages. https://doi.org/10.1145/3209900.3209911

[46] Scott Brandt, Carlos Maltzahn, Neoklis Polyzotis, and Wang-Chiew Tan. 2009. Fusing data management services with file systems. In *Proceedings of the 4th Annual Workshop on Petascale Data Storage*. ACM, 42–46.

[47] Kristin Briney. 2015. *Data Management for Researchers: Organize, maintain and share your data for research success*.

[48] Nathan Bronson, Zach Amsden, George Cabrera, Prasad Chakka, Peter Dimov, Hui Ding, Jack Ferris, Anthony Giardullo, Sachin Kulkarni, Harry Li, Mark Marchukov, Dmitri Petrov, Lovro Puzar, Yee Jiun Song, and Venkat Venkataramani. 2013. TAO: Facebook's Distributed Data Store for the Social Graph. In *Presented as part of the 2013 USENIX Annual Technical Conference (USENIX ATC*

13). USENIX, San Jose, CA, 49–60. https://www.usenix.org/conference/atc13/technical-sessions/presentation/bronson

[49] C Marlin Brown. 1999. *Human-computer interface design guidelines.* Intellect Books.

[50] Vladimir I Budzko, DA Melnikov, VI Korolev, Victor G Belenkov, and Peter A Keyer. 2019. Architecture solutions for the metadata extraction toolkit, taking into account the built-in privacy extracts. In *CEUR Workshop Proceedings*. 3–9.

[51] Vannevar Bush et al. 1945. As we may think. *The atlantic monthly* 176, 1 (1945), 101–108.

[52] Mingming Cao, Suparna Bhattacharya, and Ted Ts'o. 2007. Ext4: The Next Generation of Ext2/3 Filesystem.. In *LSF*.

[53] Lucian Carata, Sherif Akoush, Nikilesh Balakrishnan, Thomas Bytheway, Ripduman Sohan, Margo Seltzer, and Andy Hopper. 2014. A Primer on Provenance. *Commun. ACM* 57, 5 (May 2014), 52–60. https://doi.org/10.1145/2596628

[54] Neville Carvalho, Hyojun Kim, Maohua Lu, Prasenjit Sarkar, Rohit Shekhar, Tarun Thakur, Pin Zhou, Remzi H Arpaci-Dusseau, and IO Datos. 2016. Finding Consistency in an Inconsistent World: Towards Deep Semantic Understanding of Scale-out Distributed Databases.. In *HotStorage*.

[55] Peggy Cellier, M Ducassé, S Ferré, and O Ridoux. 2008. Formal concept analysis. *Lecture notes in computer science (R. Medina, S. Obiedkov, eds.)* 4933 (2008).

[56] Priya Chakriswaran, Durai Raj Vincent, Kathiravan Srinivasan, Vishal Sharma, Chuan-Yu Chang, and Daniel Gutiérrez Reina. 2019. Emotion AI-driven sentiment analysis: A survey, future research directions, and open issues. *Applied Sciences* 9, 24 (2019), 5462.

[57] Jidong Chen, Hang Guo, Wentao Wu, and Chunxin Xie. 2009. Search Your Memory ! - An Associative Memory Based Desktop Search System. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data* (Providence, Rhode Island, USA) *(SIGMOD '09)*. Association for Computing Machinery, New York, NY, USA, 1099–1102. https://doi.org/10.1145/1559845.1559992

[58] Jason Chou. 2015. FindFS - Adding Tag-Based Views to a Hierarchical Filesystem by. August (2015).

[59] Jason Chou. 2015. *FindFS: adding tag-based views to a hierarchical filesystem.* Ph. D. Dissertation. University of British Columbia.

[60] Sailesh Chutani, Owen T Anderson, Michael L Kazar, Bruce W Leverett, W Anthony Mason, Robert N Sidebotham, et al. 1992. The Episode file system. In *Proceedings of the USENIX Winter 1992 Technical Conference*. Citeseer, 43–60.

[61] Paul Hugh Cleverley and Simon Burnett. 2015. Retrieving haystacks: a data driven information needs model for faceted search. *Journal of Information Science* 41, 1 (2015), 97–113.

[62] Victor Codocedo and Amedeo Napoli. 2015. Formal Concept Analysis and Information Retrieval – A Survey. *Formal Concept Analysis: 13th International Conference, ICFCA 2015* (2015), 61–77. https://doi.org/10.1007/978-3-319-19545-2_4

[63] Anthony Collins, Trent Apted, and Judy Kay. 2007. Tabletop file system access: Associative and hierarchical approaches. In *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'07)*. IEEE, 113–120.

[64] Anthony Collins and Judy Kay. 2010. *Escaping hierarchies and desktops: Associative, pervasive file access with user control.* School of Information Technologies, University of Sydney.

[65] Matthew Conover. 2006. Analysis of the Windows Vista security model. *Symantec Advanced Threat Research* (2006).

[66] Fernando J Corbató and Victor A Vyssotsky. 1965. Introduction and overview of the Multics system. In *Proceedings of the November 30–December 1, 1965, fall joint computer conference, part I*. ACM, 185–196.

[67] Helen Custer. 1994. *Inside the Windows NT File System.*

[68] RC Daley and PG Neumann. 1965. A general-purpose file system for secondary storage. In *Proceedings of the November 30–December 1, 1965, fall joint computer conference, part I*. ACM, 213–229.

[69] Daniela Quitete de Campos Vianna. 2019. *Searching Heterogenous Personal Data.* Ph. D. Dissertation. Rutgers University.

[70] Yuri Demchenko, Canh Ngo, Cees de Laat, and Craig Lee. 2014. Federated Access Control in Heterogeneous Intercloud Environment: Basic Models and Architecture Patterns. In *2014 IEEE International Conference on Cloud Engineering*. 439–445. https://doi.org/10.1109/IC2E.2014.84

[71] David Devecsery, Michael Chow, Xianzheng Dou, Jason Flinn, and Peter M Chen. 2014. Eidetic systems. In *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*. 525–540.

[72] Daniele Di Sarli and Filippo Geraci. 2017. GFS: A Graph-Based File System Enhanced with Semantic Features. In *Proceedings of the 2017 International Conference on Information System and Data Mining* (Charleston, SC, USA) *(ICISDM '17)*. Association for Computing Machinery, New York, NY, USA, 51–55. https://doi.org/10.1145/3077584.3077591

[73] Daniele Di Sarli and Filippo Geraci. 2017. GFS: a Graph-based File System Enhanced with Semantic Features. In *Proceedings of the 2017 International Conference on Information System and Data Mining*. ACM, 51–55.

[74] Intel Digital and Xerox. 1980. The Ethernet, A Local Area Network. Data Link Layer and Physical Layer Specifications. (1980). http://decnet.ipv7.net/docs/dundas/aa-k759b-tk.pdf

[75] Edsger W Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische mathematik* 1, 1 (1959), 269–271.

[76] Paul Dourish. 2003. The Appropriation of Interactive Technologies: Some Lessons from Placeless Documents. *conference on computer supported cooperative work* 12, 4 (2003), 465–490.

[77] Paul Dourish, Richard Bentley, Rachel Jones, and Allan MacLean. 1999. Getting some perspective: using process descriptions to index document history. In *Proceedings of the international ACM SIGGROUP conference on Supporting group work*. 375–384.

[78] Paul Dourish, W. Keith Edwards, Anthony LaMarca, John Lamping, Karin Petersen, Michael Salisbury, Douglas B. Terry, and James Thornton. 2000. Extending Document Management Systems with User-Specific Active Properties. *ACM Trans. Inf. Syst.* 18, 2 (April 2000), 140–170. https://doi.org/10.1145/348751.348758

[79] Susan Dumais, Edward Cutrell, J. J. Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. 2003. Stuff I've Seen: A System for Personal Information Retrieval and Re-Use. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, Vol. 49. 28–35.

[80] Susan Dumais, Edward Cutrell, Jonathan J Cadiz, Gavin Jancke, Raman Sarin, and Daniel C Robbins. 2016. Stuff I've seen: a system for personal information retrieval and re-use. In *ACM SIGIR Forum*, Vol. 49. ACM, 28–35.

[81] Oliver Eck and Dirk Schaefer. 2011. A semantic file system for integrated product data management. *Advanced Engineering Informatics* 25, 2 (2011), 177–184. https://doi.org/10.1016/j.aei.2010.08.005

[82] R. Escriva and Emin Gun Sirer. 2015. The Design and Implementation of the WAVE Transactional File System. February (2015), 1–14. arXiv:1509.07821 http://arxiv.org/pdf/1509.07821v1.pdf

[83] Saba Eskandarian, Henry Corrigan-Gibbs, Matei Zaharia, and Dan Boneh. 2021. Express: Lowering the cost of metadata-hiding communication with cryptographic privacy. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*. 1775–1792.

[84] Marine Eviette and Andrew Simpson. 2021. Towards Models for Privacy Preservation in the Face of Metadata Exploitation. In *Privacy and Identity Management*, Michael Friedewald, Stefan Schiffner, and Stephan Krenn (Eds.). Springer International Publishing, Cham, 247–264.

[85] Michael Factor, Kalman Meth, Dalit Naor, Ohad Rodeh, and Julian Satran. 2005. Object storage: The future building block for storage systems. In *Local to Global Data Interoperability-Challenges and Technologies, 2005*. IEEE, 119–123.

[86] Sebastian Faubel and Christian Kuschel. 2008. Towards semantic file system interfaces. *CEUR Workshop Proceedings* 401 (2008).

[87] Amy Rae Fox, Arvind Satyarnarayan, Philip Guo, Haijun Xia, and James. D. Hollan. 2019. CHS: Medium: A Human-Centered Information Space:Designing Dynamic Personalized Visual Information. http://hci.ucsd.edu/220/NSF_Hollan_220.pdf

[88] Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. 2018. Cypher: An evolving query language for property graphs. In *Proceedings of the 2018 International Conference on Management of Data*. ACM, 1433–1445.

[89] Ned Freed and Dr. Nathaniel S. Borenstein. 1996. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC 2045. https://doi.org/10.17487/RFC2045

[90] Ophir Frieder and Sanjiv Kapoor. 2012. Hierarchical structured abstract data organization system. US Patent 8,209,358.

[91] Yutong Gao, Feifan Song, Xiaqing Xie, and Xuebin Gao. 2016. Implicit semantic text retrieval and distributed implementation for rural medical care. In *Cloud Computing and Intelligence Systems (CCIS), 2016 4th International Conference on*. IEEE, 264–267.

[92] Jim Gemmell, Gordon Bell, Roger Lueder, Steven Drucker, and Curtis Wong. 2002. MyLifeBits: fulfilling the Memex vision. In *Proceedings of the tenth ACM international conference on Multimedia*. 235–238.

[93] Sanjay Ghemawat, Howard Gobioff, and Shun-tak Leung. 2003. Google_File_System. (2003). https://doi.org/10.1145/1165389.945450 arXiv:z0024

[94] David Gifford, Pierre Jouvelot, Mark Sheldon, James O Toole, David K Gifford, Mark A Sheldon, and James W O Toole. 2006. 13 . Paper : Semantic File Systems Semantic File Systems. *Computer* (2006), 16–25.

[95] David K. Gifford, Pierre Jouvelot, Mark A. Sheldon, and James W. O'Toole. 1991. Semantic File Systems. In *Proceedings of the Thirteenth ACM Symposium on Operating Systems Principles* (Pacific Grove, California, USA) *(SOSP '91)*. Association for Computing Machinery, New York, NY, USA, 16–25. https://doi.org/10.1145/121132.121138

[96] David K. Gifford, Pierre Jouvelot, Mark A. Sheldon, and James W. O'Toole. 1991. Semantic File Systems. In *Proceedings of the Thirteenth ACM Symposium on Operating Systems Principles* (Pacific Grove, California, USA) *(SOSP '91)*. Association for Computing Machinery, New York, NY, USA, 16–25. https://doi.org/10.1145/121132.121138

[97] David K. Gifford, Pierre Jouvelot, Mark a. Sheldon, James W. O'Toole, Pierre Jouvelotl, Mark a. Sheldon, and W James O'Toole, Jr. 1991. Semantic file systems. *ACM SIGOPS Operating Systems Review* 25, 5 (1991), 16–25. https://doi.org/10.1145/121133.121138

[98] Joseph E Gonzalez, Reynold S Xin, Ankur Dave, Daniel Crankshaw, Michael J Franklin, and Ion Stoica. 2014. GraphX: Graph Processing in a Distributed Dataflow Framework.. In *OSDI*, Vol. 14. 599–613.

[99] Google Inc. 2019. Google Cloud Storage API - Blobs/Objects. Version 1.35.0. Online. Accessed 2021-02-01. https://googleapis.dev/python/storage/latest/blobs.html.

[100] Google Inc. 2020. Google Cloud Storage Documentation - Object Naming Guidelines. Online. Accessed 2021-02-01. https://cloud.google.com/storage/docs/naming-objects.

[299]]Gopal Burra Gopal. [n. d.]. Integrating Content-Based Access Mechanisms with Hierarchical File Systems 1 Introduction 2 The Design of the HAC File System. ([n. d.]).

[102] Burra Gopal and Udi Manber. 1999. Integrating content-based access mechanisms with hierarchical file systems. In *OSDI*, Vol. 99. 265–278.

[103] Michael Greenberg. 2021. Files-as-Filesystems for POSIX Shell Data Processing. In *Proceedings of the 11th Workshop on Programming Languages and Operating Systems* (Virtual Event, Germany) *(PLOS '21)*. Association for Computing Machinery, New York, NY, USA, 17–23. https://doi.org/10.1145/3477113.3487265

[104] Andreas Grünbacher. 2003. POSIX Access Control Lists on Linux. In *Proceedings of the FREENIX Track: 2003 USENIX Annual Technical Conference* (San Antonio, TX, USA).

[105] View User Guide. 2016. Archive. (2016), 1–2.

[106] Philip J. Guo and Margo Seltzer. 2012. BURRITO: wrapping your lab notebook in computational infrastructure. In *TaPP'12 Proceedings of the 4th USENIX conference on Theory and Practice of Provenance*. 7–7.

[107] Philip J. Guo and Margo Seltzer. 2012. BURRITO: Wrapping Your Lab Notebook in Computational Infrastructure. In *Proceedings of the 4th USENIX Workshop on the Theory and Practice of Provenance* (Boston, MA) *(TaPP'12)*. USENIX Association, Berkeley, CA, USA. http://dl.acm.org/citation.cfm?id=2342875.2342882

[108] Frank Halasz and Thomas P Moran. 1982. Analogy considered harmful. In *Proceedings of the 1982 conference on Human factors in computing systems*. ACM, 383–386.

[109] Matthew Harlan. 2011. JOIN FS : a Semantic File System for Embedded Systems. (2011).

[110] Matthew Harlan and Gabriel Parmer. 2011. JOINFS: a Semantic File System for Embedded Systems. In *Proceedings of the International Conference on Embedded Systems and Applications (ESA)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 1.

[111] Richard Harper, Siân Lindley, Eno Thereska, Richard Banks, Philip Gosset, Gavin Smyth, William Odom, and Eryn Whitworth. 2013. What is a File?. In *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 1125–1136.

[112] Steve Harrison and Paul Dourish. 1996. Re-place-ing space: the roles of place and space in collaborative systems. In *Proceedings of the 1996 ACM conference on Computer supported cooperative work*. 67–76.

[113] Marti Hearst. 2006. Design recommendations for hierarchical faceted search interfaces. In *ACM SIGIR workshop on faceted search*. Seattle, WA, 1–5.

[114] Joseph M Hellerstein, Vikram Sreekanti, Joseph E Gonzalez, James Dalton, Akon Dey, Sreyashi Nag, Krishna Ramachandran, Sudhanshu Arora, Arka Bhattacharyya, Shirshanka Das, et al. 2017. Ground: A Data Context Service.. In *CIDR*. Citeseer.

[115] John H. Howard, Michael L. Kazar, Sherri G. Menees, David A. Nichols, M. Satyanarayanan, Robert N. Sidebotham, and Michael J. West. 1988. Scale and performance in a distributed file system. *ACM Transactions on Computer Systems* 6, 1 (1988), 51–81.

[116] Yu Hua, Hong Jiang, and Dan Feng. 2016. Real-time semantic search using approximate methodology for large-scale storage systems. *IEEE Transactions on Parallel and Distributed Systems* 27, 4 (2016), 1212–1225.

[117] Yu Hua, Hong Jiang, Yifeng Zhu, and Dan Feng. 2010. Rapport: Semantic-sensitive Namespace Management in Large-scale File Systems. *CSE Technical reports* (2010). http://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1123{&}context=csetechreports

[118] Yu Hua, Hong Jiang, Yifeng Zhu, Dan Feng, and Lei Tian. 2009. SmartStore: a new metadata organization paradigm with semantic-awareness for next-generation file systems. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*. 1–12. https://doi.org/10.1145/1654059.1654070

[119] H Howie Huang, Nan Zhang, Wei Wang, Gautam Das, and Alexander S Szalay. 2012. Just-in-time analytics on large file systems. *IEEE Trans. Comput.* 61, 11 (2012), 1651–1664.

[120] Jun Huang and ShunXiang Wu. 2012. The research of fast file search engine based on NTFS and its application in fast electronic document destruction. In *Advances in Computer Science and Information Engineering*. Springer, 523–528.

[121] HUAWEI TECHNOLOGIES CO., LTD. 2020. Multi-Cloud Container Platform: Namespaces. Issue 01. Online. Accessed 2021-02-01. https://support.huaweicloud.com/intl/en-us/usermanual-mcp/mcp_01_0025.html.

[122] Zhisheng Huo, Limin Xiao, Qiaoling Zhong, Shupan Li, Ang Li, Li Ruan, Shouxin Wang, and Lihong Fu. 2016. MBFS: a parallel metadata search method based on Bloomfilters using MapReduce for large-scale file systems. *The Journal of Supercomputing* 72, 8 (2016), 3006–3032.

[123] Hugo C Huurdeman, Max L Wilson, and Jaap Kamps. 2016. Active and passive utility of search interface features in different information seeking task stages. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*. ACM, 3–12.

[124] Shelby Louise Hyvonen and Shelby Louise. 2004. Reproduced with permission of the copyright owner. Further reproduction prohibited without permission. *Dissertation* May (2004).

[125] IBM Corporation. 2020. WebSphere Application Server (Distributed operating systems) Namespace Federation. Version 8.5.5. Online. Accessed 2021-02-01. https://www.ibm.com/support/knowledgecenter/SSEQTP_8.5.5/com.ibm.websphere.base.doc/ae/cnam_federation.html.

[126] Mohammad Jahanian, Jiachen Chen, and K. K. Ramakrishnan. 2021. Graph-Based Namespaces and Load Sharing for Efficient Information Dissemination. *IEEE/ACM Transactions on Networking* (2021), 1–14. https://doi.org/10.1109/TNET.2021.3094839

[127] Sadaqat Jan, Maozhen Li, Ghaidaa Al-Sultany, Hamed Al-Raweshidy, and Ibrar Ali Shah. 2011. Semantic file annotation and retrieval on mobile devices. *Mobile Information Systems* 7, 2 (2011), 107–122. https://doi.org/10.3233/MIS-2011-0113

[128] TS Jayalakshmi and C Chethana. 2016. A semantic search engine for indexing and retrieval of relevant text documents. (2016).

[129] Carlos Jensen, Heather Lonsdale, Eleanor Wynn, Jill Cao, Michael Slater, and Thomas G Dietterich. 2010. The life and times of files and information: a study of desktop provenance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 767–776.

[130] ROBIN JESHION. 2009. The Significance of Names. *Mind & Language* 24, 4 (2009), 370–403. https://doi.org/10.1111/j.1468-0017.2009.01367.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1468-0017.2009.01367.x

[131] Indiana Jones, What You, and C A N Do. 2016. Tagxfs is a semantic file system. It extends the user space file system to a tag based hierarchy. (2016), 1–5.

[132] William Jones, Ammy Jiranida Phuwanartnurak, Rajdeep Gill, and Harry Bruce. 2005. Don't take my folders away!: organizing personal information to get things done. In *CHI'05 extended abstracts on Human factors in computing systems*. ACM, 1505–1508.

[299]]Joshi Amruta Joshi. [n. d.]. Orion File System : File-level Host-based Virtualization. ([n. d.]).

[134] Rohan Kadekodi, Saurabh Kadekodi, Soujanya Ponnapalli, Harshad Shirwadkar, Gregory R Ganger, Aasheesh Kolli, and Vijay Chidambaram. 2021. WineFS: a hugepage-aware file system for persistent memory that ages gracefully. In *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles CD-ROM*. 804–818.

[135] Varvara Kalokyri, Alexander Borgida, Amélie Marian, and Daniela Vianna. 2017. Integration and Exploration of Connected Personal Digital Traces. In *Proceedings of the ExploreDB'17*. 3.

[136] Hyunmo Kang and Ben Shneiderman. 2003. MediaFinder: An Interface for Dynamic Personal Media Management with Semantic Regions. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI EA '03)*. Association for Computing Machinery, New York, NY, USA, 764–765. https://doi.org/10.1145/765891.765977

[137] David R Karger and William Jones. 2006. Data unification in personal information management. *Commun. ACM* 49, 1 (2006), 77–82.

[138] Amy K Karlson, Greg Smith, and Bongshin Lee. 2011. Which version is this?: improving the desktop experience within a copy-aware computing ecosystem. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2669–2678.

[139] Dmitry Kasatkin and Zohar Miriam. 2021. Integrity Measurement Architecture (IMA). (2021). https://sourceforge.net/p/linux-ima/wiki/Home/

[140] Aditya Kashyap and A Kashyap. 2004. *File system extensibility and reliability using an in-kernel database*. Ph. D. Dissertation. Stony Brook University.

[141] Yutaka Kawai, Adil Hasan, Go Iwai, Takashi Sasaki, and Yoshiyuki Watase. 2011. A method for reliably managing files with RNS in multi Data Grids. In *Procedia Computer Science*, Vol. 4. 412–421.

[142] Michael L. Kazar, Bruce W. Leverett, Owen T. Anderson, Vasilis Apostolides, Beth A. Bottos, Sailesh Chutani, Craig Everhart, W. Anthony Mason, Shu-Tsui Tu, and Edward R. Zayas. 1990. DEcorum File System Architectural Overview. *USENIX Summer* (1990), 151–164.

[143] Mohammad Taha Khan, Maria Hyun, Chris Kanich, and Blase Ur. 2018. Forgotten But Not Gone: Identifying the Need for Longitudinal Data Management in Cloud Storage. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 543.

[144] Jinyoung Kim and W. Bruce Croft. 2009. Retrieval experiments using pseudo-desktop collections. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 1297–1306.

[145] Vidar Klungre and Martin Giese. 2018. Evaluating a Faceted Search Index for Graph Data. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 573–583.

[146] John Koetsier. 2013. Digg pokes up its head and says, 'I'm not dead yet'. https://tinyurl.com/yam3xyp4 (accessed January 6, 2019).

[147] Jinhyung Koo, Junsu Im, Jooyoung Song, Juhyung Park, Eunji Lee, Bryan S. Kim, and Sungjin Lee. 2021. Modernizing File System through In-Storage Indexing. In *15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21)*. 75–92.

[148] Jonathan Koren, Andrew Leung, Yi Zhang, Carlos Maltzahn, Sasha Ames, and Ethan Miller. 2007. Searching and navigating petabyte-scale file systems based on facets. In *Proceedings of the 2nd international workshop on Petascale data storage: held in conjunction with Supercomputing'07*. ACM, 21–25.

[149] Harsha P Kumar, Catherine Plaisant, and Ben Shneiderman. 1997. Browsing hierarchical data with multi-level dynamic queries and pruning. *International journal of human-computer studies* 46, 1 (1997), 103–124.

[150] Aapo Kyrola, Guy E Blelloch, and Carlos Guestrin. 2012. Graphchi: Large-scale graph computation on just a pc. USENIX.

[151] Keith A Lantz, Judy L Edighoffer, and Bruce L Hitson. 1986. Towards a Universal Directory Service. *SIGOPS Oper. Syst. Rev.* 20, 2 (April 1986), 43–53. https://doi.org/10.1145/12481.12483

[152] Aaron Laursen. 2014. *A Novel, Tag-based File-System*. Master's thesis. Macalester College. https://digitalcommons.macalester.edu/mathcs_honors/34/ (accessed March 29, 2018).

[153] Changman Lee, Dongho Sim, Jooyoung Hwang, and Sangyeun Cho. 2015. F2FS: A new file system for flash storage. In *13th {USENIX} Conference on File and Storage Technologies ({FAST} 15)*. 273–286.

[154] Paul Hermann Lensing, Toni Cortes, and André Brinkmann. 2013. Direct Lookup and Hash-Based Metadata Placement for Local File Systems. In *Proceedings of the 6th International Systems and Storage Conference* (Haifa, Israel) *(SYSTOR '13)*. Association for Computing Machinery, New York, NY, USA, Article 5, 11 pages. https://doi.org/10.1145/2485732.2485741

[155] Andrew Leung, I Adams, and Ethan L Miller. 2009. Magellan: A searchable metadata architecture for large-scale file systems. *University of California, Santa Cruz, Tech. Rep. UCSC-SSRC-09-07* (2009).

[156] A Leung, A Parker-Wood, and EL Miller. 2009. Copernicus: A scalable, high-performance semantic file system. *University of California, Santa ...* October (2009). http://www.ssrc.ucsc.edu/papers/ssrctr-09-06.pdf

[157] Andrew W Leung, Minglong Shao, Timothy Bisson, Shankar Pasupathy, and Ethan L Miller. 2009. Spyglass: Fast, Scalable Metadata Search for Large-Scale Storage Systems.. In *FAST*, Vol. 9. 153–166.

[158] Roy Levin and Michael D Schroeder. 1979. *Transport of electronic messages through a network.* Technical Report. Xerox. Palo Alto Research Center.

[159] Yan Li, Nakul Sanjay Dhotre, Yasuhiro Ohara, Thomas M. Kroeger, Ethan L. Miller, and Darrell D. E. Long. 2013. Horus: fine-grained encryption-based security for large-scale storage. In *Proceedings of the 11th USENIX conference on File and Storage Technologies*. 147–160.

[160] Yujie Li, Yinging Jiang, Daxin Tian, Long Hu, Huimin Lu, and Zhiyong Yuan. 2019. AI-Enabled Emotion Communication. *IEEE Network* 33, 6 (2019), 15–21. https://doi.org/10.1109/MNET.001.1900070

[161] Hyeontaek Lim, Bin Fan, David G Andersen, and Michael Kaminsky. 2011. SILT: A memory-efficient, high-performance key-value store. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*. ACM, 1–13.

[162] Siân E Lindley, Gavin Smyth, Robert Corish, Anastasia Loukianov, Michael Golembewski, Ewa A Luger, and Abigail Sellen. 2018. Exploring New Metaphors for a Networked World through the File Biography. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 118.

[163] Gentoo Linux. 2020. How SELinux controls file and directory accesses. (2020). https://wiki.gentoo.org/wiki/SELinux/Tutorials/How_SELinux_controls_file_and_directory_accesses

[164] Linux man-pages Project. 2020. *xattr - Extended attributes.* Linux Programmer's Manual, Online. Accessed 2021-02-01. https://man7.org/linux/man-pages/man7/xattr.7.html.

[165] Jinjun Liu, Dan Feng, Yu Hua, Bin Peng, Pengfei Zuo, and Yuanyuan Sun. 2015. P-index: An efficient searchable metadata indexing scheme based on data provenance in cold storage. In *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 597–611.

[166] Xiaotao Liu, Gal Niv, Prashant Shenoy, K.K. Ramakrishnan, and Jacobus Van der Merwe. 2006. The case for semantic aware remote replication. *Proceedings of the second ACM workshop on Storage security and survivability - StorageSS '06* (2006), 79. https://doi.org/10.1145/1179559.1179575

[167] John Locke. 1844. *Locke's essays: An essay concerning human understanding, and A treatise on the conduct of the understanding (Complete in 1 volume with the author's last additions and corrections).*

[168] Yucheng Low, Joseph E Gonzalez, Aapo Kyrola, Danny Bickson, Carlos E Guestrin, and Joseph Hellerstein. 2014. Graphlab: A new framework for parallel machine learning. *arXiv preprint arXiv:1408.2041* (2014).

[169] Shanshan Ma and Susan Wiedenbeck. 2009. File management with hierarchical folders and tags. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 3745–3750.

[170] John MacCormick, Nick Murphy, Marc Najork, Chandramohan A Thekkath, and Lidong Zhou. 2004. Boxwood: Abstractions as the Foundation for Storage Infrastructure.. In *OSDI*, Vol. 4. 8–8.

[171] DN MacDonald. 1956. Datafile: a new tool for extensive file storage. In *Papers and discussions presented at the December 10-12, 1956, eastern joint computer conference: New developments in computers*. ACM, 124–128.

[172] Peter Macko, Virendra J Marathe, Daniel Wyatt Margo, and Margo I Seltzer. 2015. LLAMA: Efficient graph analytics using large multiversioned arrays. In *Proceedings of the 31st IEEE International Conference on Data Engineering*. IEEE.

[173] Peter Macko, Daniel Margo, and Margo Seltzer. 2013. Performance introspection of graph databases. In *Proceedings of the 6th International Systems and Storage Conference*. ACM, 18.

[174] M. Mahalingam, Chunqiang Tang, and Zhichen Xu. 2003. Towards a semantic, deep archival file system. *Proceedings of the IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems* 2003-Janua (2003), 115–121. https://doi.org/10.1109/FTDCS.2003.1204321

[175] Grzegorz Malewicz, Matthew H Austern, Aart JC Bik, James C Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. 2010. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 135–146.

[176] Thomas W Malone. 1983. How do people organize their desks?: Implications for the design of office information systems. *ACM Transactions on Information Systems (TOIS)* 1, 1 (1983), 99–112.

[177] Thomas W. Malone. 1983. How do people organize their desks?: Implications for the design of office information systems. *ACM Transactions on Information Systems* 1, 1 (1983), 99–112.

[178] Richard Mander, Gitta Salomon, and Yin Yin Wong. 1992. A "Pile" Metaphor for Supporting Casual Organization of Information. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '92* (1992), 627–634. https://doi.org/10.1145/142750.143055

[179] Yandong Mao, Eddie Kohler, and Robert Tappan Morris. 2012. Cache craftiness for fast multicore key-value storage. In *Proceedings of the 7th ACM european conference on Computer Systems*. ACM, 183–196.

[180] Gary Marchionini. 2006. Exploratory search: from finding to understanding. *Commun. ACM* 49, 4 (2006), 41–46.

[181] Daniel Margo and Margo Seltzer. 2015. A scalable distributed graph partitioner. *Proceedings of the VLDB Endowment* 8, 12 (2015), 1478–1489.

[182] Gary Marsden and David E Cairns. 2003. Improving the usability of the hierarchical file system. In *Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*. South African Institute for Computer Scientists and Information Technologists, 122–129.

[183] Ben Martin. 2004. Formal concept analysis and semantic file systems. In *International Conference on Formal Concept Analysis*. Springer, 88–95.

[184] B Martin. 2004. Formal concept analysis and semantic file systems. *Concept Lattices* (2004). http://link.springer.com/chapter/10.1007/978-3-540-24651-0{_}9

[185] Ben Martin. 2008. *Formal concept analysis and semantic file systems.* Ph.D. Dissertation. University of Wollongong. https://ro.uow.edu.au/theses/260/

[186] Ben Martin. 2014. Open Source Libferris: Chasing the "Everything is a File System" Dream. *linux.com* (23 January 2014). https://www.linux.com/learn/open-source-libferris-chasing-everything-file-system-dream (accessed September 4, 2018).

[187] B. Martin and P. Eklund. 2005. Applying formal concept analysis to semantic file systems leveraging wordnet. *ADCS 2005 - Proceedings of the Tenth Australasian Document Computing Symposium* (2005), 56–63.

[188] Jonathan Masci, Michael M Bronstein, Alexander M Bronstein, and Jürgen Schmidhuber. 2014. Multimodal similarity-preserving hashing. *IEEE transactions on pattern analysis and machine intelligence* 36, 4 (2014), 824–830.

[189] Syed Rahman Mashwani and Shah Khusro. 2019. 360° Semantic File System: Augmented Directory Navigation for Nonhierarchical Retrieval of Files. *IEEE Access* 7 (2019), 9406–9418.

[190] Michelle L. Mazurek, Yuan Liang, William Melicher, Manya Sleeper, Lujo Bauer, Gregory R. Ganger, Nitin Gupta, and Michael K. Reiter. 2014. Toward strong, usable access control for shared distributed data. In *Proceedings of the 12th USENIX conference on File and Storage Technologies*. 89–103.

[191] Marshall K. McKusick, William N. Joy, Samuel J. Leffler, and Robert S. Fabry. 1984. A fast file system for UNIX. *ACM Transactions on Computer Systems* 2, 3 (1984), 181–197.

[192] Marshall K McKusick, William N Joy, Samuel J Leffler, and Robert S Fabry. 1984. A fast file system for UNIX. *ACM Transactions on Computer Systems (TOCS)* 2, 3 (1984), 181–197.

[193] Eva Méndez and Jane Greenberg. 2012. Linked data for open vocabularies and HIVE's global framework. *El profesional de la información* 21, 3 (2012), 236–244.
[299]]microsoft:data-add-in Microsoft. [n. d.]. Data Add-In Interfaces. https://tinyurl.com/y83elc34 (accessed January 6, 2019).
[195] Microsoft. 2019. Welcome to Azure Cosmos DB. https://docs.microsoft.com/en-us/azure/cosmos-db/introduction (accessed January 17, 2019).
[196] Changwoo Min, Sanidhya Kashyap, Byoungyoung Lee, Chengyu Song, and Taesoo Kim. 2015. Cross-checking Semantic Correctness : The Case of Finding File System Bugs. *Sosp '15* (2015), 361–377. https://doi.org/10.1145/2815400.2815422
[197] Jeffrey Clifford Mogul. 1986. *Representing information about files.* Ph. D. Dissertation. Stanford University.
[198] James H. Morris, Mahadev Satyanarayanan, Michael H. Conner, John H. Howard, David S. Rosenthal, and F. Donelson Smith. 1986. Andrew: A Distributed Personal Computing Environment. *Commun. ACM* 29, 3 (March 1986), 184–201. https://doi.org/10.1145/5666.5671
[199] Kiran-Kumar Muniswamy-Reddy, David A. Holland, Uri Braun, and Margo Seltzer. 2006. Provenance-aware Storage Systems. In *Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference* (Boston, MA) *(ATEC '06)*. USENIX Association, Berkeley, CA, USA, 4–4. http://dl.acm.org/citation.cfm?id=1267359.1267363
[200] Subramanian Muralidhar, Wyatt Lloyd, Sabyasachi Roy, Cory Hill, Ernest Lin, Weiwen Liu, Satadru Pan, Shiva Shankar, Viswanath Sivakumar, Linpeng Tang, et al. 2014. f4: Facebook's warm blob storage system. In *Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation*. USENIX Association, 383–398.
[201] Daniel L. Murphy. 1972. Storage organization and management in TENEX. In *Proceedings of the December 5-7, 1972, fall joint computer conference, part I*. 23–32.
[202] Nick Murphy, Mark Tonkelowitz, and Mike Vernal. 2002. The design and implementation of the database file system.
[203] nayuki. 2017. Designing better file organization around tags, not hierarchies. https://www.nayuki.io/page/designing-better-file-organization-around-tags-not-hierarchies (accessed October 1, 2018).
[204] Roger M Needham and Andrew D Birrell. 1977. The CAP filing system. In *ACM SIGOPS Operating Systems Review*, Vol. 11. ACM, 11–16.
[205] T. H. Nelson. 1965. 4.2: A File Structure for The Complex, The Changing and the Indeterminate. (1965).
[206] Ba-hung Ngo, Christian Bac, Ba-hung Ngo, Christian Bac, and Frédérique Silber-chaussumier. 2015. To cite this version : Integrating Ontology into Semantic File Systems. (2015).
[207] Ba-Hung Ngo, Christian Bac, and Frédérique Silber-Chaussumier. 2007. Integrating ontology into semantic file systems. In *Huitièmes Journées Doctorales en Informatique et Réseaux (JDIR'07)*. 139–142.
[208] Donald Nguyen, Andrew Lenharth, and Keshav Pingali. 2013. A lightweight infrastructure for graph analytics. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*. ACM, 456–471.
[209] Salman Niazi, Mahmoud Ismail, Seif Haridi, Jim Dowling, Steffen Grohsschmiedt, and Mikael Ronström. 2017. HopsFS: Scaling Hierarchical File System Metadata Using NewSQL Databases.. In *FAST*. 89–104.
[210] R. Noronha and D.K. Panda. 2008. IMCa: A High Performance Caching Front-End for GlusterFS on InfiniBand. In *2008 37th International Conference on Parallel Processing*. 462–469.
[211] Object Management Group, Inc. 2004. Naming Service Specification. Version 1.3. Online. Accessed 2021-02-01. https://www.omg.org/spec/NAM/1.3/PDF.
[212] William Odom, Abi Sellen, Richard Harper, and Eno Thereska. 2012. Lost in translation: understanding the possession of digital things in the cloud. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 781–790.
[213] Michael A. Olson. 1993. The Design and Implementation of the Inversion File System. In *USENIX Winter 1993 Conference (USENIX Winter 1993 Conference)*. USENIX Association, San Diego, CA. https://www.usenix.org/conference/usenix-winter-1993-conference/design-and-implementation-inversion-file-system
[214] Michael A Olson et al. 1993. The Design and Implementation of the Inversion File System. In *USENIX Winter*. 205–218.
[215] Pieter Omvlee. 2009. A novel idea for a new Filesystem. *June* (2009). http://referaat.cs.utwente.nl/conference/11/paper/6966/a-novel-idea-for-a-new-filesystem.pdf
[216] Personal Open and Business Explore. 2016. Fuse::TagLayer. (2016), 1–2.
[217] Oracle. 2020. Java Platform Standard Ed. 7 - Package org.omg.CosNaming. Online. Accessed 2021-02-01. https://docs.oracle.com/javase/7/docs/api/org/omg/CosNaming/package-summary.html.
[218] Laurel Orr, Magda Balazinska, and Dan Suciu. 2020. Sample Debiasing in the Themis Open World Database System (Extended Version). arXiv:2002.09799 [cs.DB]
[219] Mac Os, Overview Xtagfs, Mac Os, Spotlight Comment, Mac Os, Tagging Files Files, Spotlight Comments, Get Info, Spotlight Comments, Install Macfuse, Download Xtagfs, and Run Xtagfs. 2016. Archive. (2016), 1–2.
[220] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.
[221] Aleatha Parker-Wood, Darrell D E Long, Ethan Miller, Philippe Rigaux, and Andy Isaacson. 2014. A File By Any Other Name: Managing File Names with Metadata. *Proceedings of International Conference on Systems and Storage* (2014), 1–11.
[222] Aleatha Parker-Wood, Darrell D. E. Long, Ethan Miller, Philippe Rigaux, and Andy Isaacson. 2014. A File By Any Other Name: Managing File Names with Metadata. In *Proceedings of International Conference on Systems and Storage* (Haifa, Israel) *(SYSTOR 2014)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/2611354.2611367
[223] Jack Parkinson and Quintin Cutts. 2018. Investigating the Relationship Between Spatial Skills and Computer Science. 106–114. https://doi.org/10.1145/3230977.3230990
[224] Thomas Pasquier, Xueyuan Han, Mark Goldstein, Thomas Moyer, David Eyers, Margo Seltzer, and Jean Bacon. 2017. Practical Whole-System Provenance Capture. In *Symposium on Cloud Computing (SoCC'17)*. ACM.
[225] Thomas Pasquier, Xueyuan Han, Mark Goldstein, Thomas Moyer, David Eyers, Margo Seltzer, and Jean Bacon. 2017. Practical Whole-system Provenance Capture. In *Proceedings of the 2017 Symposium on Cloud Computing* (Santa Clara, California) *(SoCC '17)*. ACM, New York, NY, USA, 405–418. https://doi.org/10.1145/3127479.3129249
[226] Rob Pike, Dave Presotto, Ken Thompson, Howard Trickey, and Phil Winterbottom. 1992. The use of name spaces in Plan 9. In *Proceedings of the 5th workshop on ACM SIGOPS European workshop: Models and paradigms for distributed systems structuring*. ACM, 1–5.
[227] Rob Pike, Dave Presotto, Ken Thompson, Howard Trickey, and Phil Winterbottom. 1993. The Use of Name Spaces in Plan 9. *SIGOPS Oper. Syst. Rev.* 27, 2 (April 1993), 72–76. https://doi.org/10.1145/155848.155861
[228] Rob Pike, Dave Presotto, Ken Thompson, Howard Trickey, and Phil Winterbottom. 1993. The Use of Name Spaces in Plan 9. *SIGOPS Oper. Syst. Rev.* 27, 2 (April 1993), 72–76. https://doi.org/10.1145/155848.155861
[229] Rob Pike and Peter Weinberger. 1985. The Hideous Name. In *USENIX Summer 1985 Conference Proceedings* (Portland Oregon).
[230] Vijayan Prabhakaran, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau. 2005. Analysis and evolution of journaling file systems. *Proceedings of the International Conference on Dependable Systems and Networks* (2005), 8. https://doi.org/10.1109/DSN.2005.65
[231] Dale Rahn. 2005. CVS log for src/sys/ufs/ufs/Attic/extattr.h. Revision 1.2. Online. Accessed 2021-02-01. https://cvsweb.openbsd.org/src/sys/ufs/ufs/Attic/extattr.h.
[232] Annajiat Alim Rasel and Mohammed Eunus Ali. 2016. UProve2: privacy-aware, scalable, ubiquitous provenance to enhance file search. In *2016 International Conference on Networking Systems and Security (NSysS)*. IEEE, 1–5.
[233] David Reinsel, John Gantz, and John Rydning. 2018. *Data Age 2025: The Digitization of the World From Edge to Core.* Technical Report. Seagate. https://tinyurl.com/y95ogat4 (accessed January 6, 2019).
[234] Hans Reiser. 2007. Name Spaces as Tools for Integrating the Operating System Rather Than As Ends in Themselves. (2007). http://www.namesys.com/whitepaper.html (accessed October 28, 2016 via archive.org).
[235] Kai Ren and Garth A Gibson. 2013. TABLEFS: Enhancing Metadata Efficiency in the Local File System.. In *USENIX Annual Technical Conference*. 145–156.
[236] Francois Revol. 2011. Universal file system extended attributes namespace. In *International Conference on Dublin Core and Metadata Applications*. 69–73.
[237] Dennis M Ritchie and Ken Thompson. 1973. The UNIX time-sharing system. In *ACM SIGOPS Operating Systems Review*, Vol. 7. ACM, 27.
[238] Dennis M. Ritchie and Ken Thompson. 1974. The UNIX Time-Sharing System. *Commun. ACM* 17, 7 (July 1974), 365–375. https://doi.org/10.1145/361011.361061
[239] Hubert Ritzdorf, Nikolaos Karapanos, and Srdjan Čapkun. 2014. Assisted deletion of related content. In *Proceedings of the 30th Annual Computer Security Applications Conference*. 206–215.
[240] Kerry Rodden and Kenneth R. Wood. 2003. How Do People Manage Their Digital Photographs?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI '03)*. Association for Computing Machinery, New York, NY, USA, 409–416. https://doi.org/10.1145/642611.642682
[241] Mendel Rosenblum and John K Ousterhout. 1992. The design and implementation of a log-structured file system. *ACM Transactions on Computer Systems (TOCS)* 10, 1 (1992), 26–52.
[242] Louis Rosenfeld and Peter Morville. 2002. *Information architecture for the world wide web.* " O'Reilly Media, Inc.".
[243] Michael Rudolf, Marcus Paradies, Christof Bornhövd, and Wolfgang Lehner. 2013. The Graph Story of the SAP HANA Database.. In *BTW*, Vol. 13. Citeseer, 403–420.

[244] Semih Salihoglu and Jennifer Widom. 2013. GPS: a graph processing system. In *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*. ACM, 22.

[245] Jerome H. Saltzer. 1978. *Naming and binding of objects*. Springer Berlin Heidelberg, Berlin, Heidelberg, 99–208. https://doi.org/10.1007/3-540-08755-9_4

[246] Russel Sandberg. 1986. The Sun network file system: Design, implementation and experience. In *in Proceedings of the Summer 1986 USENIX Technical Conference and Exhibition*. Citeseer.

[247] Russel Sandberg, David Goldberg, Steve Kleiman, Dan Walsh, and Bob Lyon. 1985. Design and implementation of the Sun network filesystem. In *Proceedings of the Summer USENIX conference*. 119–130.

[248] Bernhard Schandl. 2009. Representing linked data as virtual file systems. *CEUR Workshop Proceedings* 538 (2009).

[249] Bernhard Schandl and Bernhard Haslhofer. 2009. The sile model - A semantic file system infrastructure for the desktop. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5554 LNCS (2009), 51–65. https://doi.org/10.1007/978-3-642-02121-3_8

[250] Margo Seltzer and Nicholas Murphy. 2009. Hierarchical file systems are dead. In *HotOS'09 Proceedings of the 12th conference on Hot topics in operating systems*. 1–1.

[251] Margo Seltzer and Nicholas Murphy. 2009. Hierarchical File Systems Are Dead. In *Proceedings of the 12th Conference on Hot Topics in Operating Systems* (Monte Verità, Switzerland) *(HotOS'09)*. USENIX Association, USA, 1.

[252] Abhey Shah. 2007. Knowledge Management Enviroments for High Throughput Biology. September (2007).

[253] Abkey Shah and Leo Caves. 2006. ConceptOntoFs : A Semantic File System for Inferno. *1st International Workshop on Plan 9* (2006).

[254] Sam Shah, Craig A. N. Soules, Gregory R. Ganger, and Brian D. Noble. 2007. Using Provenance to Aid in Personal File Search. In *2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference* (Santa Clara, CA) *(ATC'07)*. USENIX Association, USA, Article 13, 14 pages.

[255] Jonathan S Shapiro. 2004. extracting the lessons of Multics. (2004).

[256] Pradeep J. Shetty, Richard P. Spillane, Ravikant R. Malpani, Binesh Andrews, Justin Seyster, and Erez Zadok. 2013. Building Workload-Independent Storage with VT-Trees. In *Presented as part of the 11th USENIX Conference on File and Storage Technologies (FAST 13)*. USENIX, San Jose, CA, 17–30. https://www.usenix.org/conference/fast13/technical-sessions/presentation/shetty

[257] Julian Shun and Guy E Blelloch. 2013. Ligra: a lightweight graph processing framework for shared memory. In *ACM Sigplan Notices*, Vol. 48. ACM, 135–146.

[258] Bob Sidebotham et al. 1986. *Volumes: the andrew file system data structuring primitive*. Carnegie Mellon University, Information Technology Center.

[299]]Siekmann J Siekmann. [n. d.]. Formal Concept Analysis. ([n. d.]), 0–45.

[260] H. Sim, A. Khan, S. S. Vazhkudai, S. Lim, A. R. Butt, and Y. Kim. 2020. An Integrated Indexing and Search Service for Distributed File Systems. *IEEE Transactions on Parallel and Distributed Systems* 31, 10 (2020), 2375–2391. https://doi.org/10.1109/TPDS.2020.2990656

[261] Swaminathan Sivasubramanian. 2012. Amazon DynamoDB: A Seamlessly Scalable Non-Relational Database Service. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data* (Scottsdale, Arizona, USA) *(SIGMOD '12)*. Association for Computing Machinery, New York, NY, USA, 729–730. https://doi.org/10.1145/2213836.2213945

[262] CAN Soules and GR Ganger. 2004. Toward automatic context-based attribute assignment for semantic file systems. *Parallel data laboratory, Carnegie Mellon ...* June (2004). http://shiftleft.com/mirrors/www.hpl.hp.com/personal/Craig{_}Soules/papers/CMU-PDL-04-105.pdf

[263] Craig AN Soules and Gregory R Ganger. 2003. Why Can't I Find My Files?: New Methods for Automating Attribute Assignment. In *HotOS*. 115–120.

[264] Craig AN Soules and Gregory R Ganger. 2005. Connections: using context to enhance file search. In *Proceedings of the twentieth ACM symposium on operating systems principles*. 119–132.

[265] Alfred Z. Spector. 1981. Performing Remote Operations Efficiently on a Local Computer Network. *SIGOPS Oper. Syst. Rev.* 15, 5 (Dec. 1981), 76–77. https://doi.org/10.1145/1067627.806594

[266] Alfred Z. Spector. 1981. Performing Remote Operations Efficiently on a Local Computer Network. In *Proceedings of the Eighth ACM Symposium on Operating Systems Principles* (Pacific Grove, California, USA) *(SOSP '81)*. Association for Computing Machinery, New York, NY, USA, 76–77. https://doi.org/10.1145/800216.806594

[267] Michale J. Spier, Thomas N. Hastings, and David N. Cutler. 1973. An Experimental Implementation of the Kernel/Domain Architecture. In *Proceedings of the Fourth ACM Symposium on Operating System Principles (SOSP '73)*. ACM, New York, NY, USA, 8–21. https://doi.org/10.1145/800009.808043

[268] Mahesh K Sreenivas, Stefan R Steiner, Arkadi Brjazovski, and Sameet H Agarwal. 2011. Bypass of the namespace hierarchy to open files. US Patent 7,925,681.

[269] Suresh Srinivas. 2011. An Introduction to HDFS Federation. Online. Accessed 2021-02-01. https://blog.cloudera.com/an-introduction-to-hdfs-federation/.

[270] Michael Stonebraker, Paul M Aoki, Robert Devine, Witold Litwin, and Michael Olson. 1994. Mariposa: A new architecture for distributed data. In *Data Engineering, 1994. Proceedings. 10th International Conference*. IEEE, 54–65.

[299]]Strong Tyronda Strong and Pavani Akundi. [n. d.]. A Semantic Cloud for File System Annotation. ([n. d.]), 1–4.

[272] Tyronda Strong and Pavani Akundi. 2013. A Semantic Cloud for File System Annotation. In *Proceedings of the International Conference on Semantic Web and Web Services (SWWS)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 24.

[273] M Suguna and T Anand. 2015. Dynamic Metadata Management in Semantic File Systems. 5, 3 (2015), 44–47.

[274] Ignacio G Terrizzano, Peter M Schwarz, Mary Roth, and John E Colino. 2015. Data Wrangling: The Challenging Yourney from the Wild to the Lake.. In *CIDR*.

[275] The Kubernetes Authors. 2021. Kubernetes Concepts: Namespaces. Version 1.16. Online. Accessed 2021-02-01. https://v1-16.docs.kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/.

[276] Eno Thereska, Oriana Riva, Richard Banks, Sian Lindley, Richard Harper, and William Odom. 2013. *Beyond file systems: understanding the nature of places where people store their data*. Technical Report. Microsoft Research. https://www.microsoft.com/en-us/research/wp-content/uploads/2013/02/MSR-TR-2013-26.pdf (accessed March 29, 2018).

[277] Marco Todesco, Gregory L. Owens, Natalia Bercovich, Jean-Sébastien Légaré, Shaghayegh Soudi, Dylan O. Burge, Kaichi Huang, Katherine L. Ostevik, Emily B. M. Drummond, Ivana Imerovski, Kathryn Lande, Mariana A. Pascual-Robles, Mihir Nanavati, Mojtaba Jahani, Winnie Cheung, S. Evan Staton, Stéphane Muños, Rasmus Nielsen, Lisa A. Donovan, John M. Burke, Sam Yeaman, and Loren H. Rieseberg. 2020. Massive haplotypes underlie ecotypic differentiation in sunflowers. *Nature* 584, 7822 (2020), 602–607.

[278] Daniel Tunkelang. 2009. Faceted search. *Synthesis lectures on information concepts, retrieval, and services* 1, 1 (2009), 1–80.

[279] Las Up. 2016. Tag , no di ! (2016), 1–23.

[280] Dana Van Aken, Andrew Pavlo, Geoffrey J. Gordon, and Bohan Zhang. 2017. Automatic Database Management System Tuning Through Large-Scale Machine Learning. In *Proceedings of the 2017 ACM International Conference on Management of Data* (Chicago, Illinois, USA) *(SIGMOD '17)*. Association for Computing Machinery, New York, NY, USA, 1009–1024. https://doi.org/10.1145/3035918.3064029

[281] Frank Van Ham and Jarke J van Wijk. 2003. Beamtrees: Compact visualization of large hierarchies. *Information Visualization* 2, 1 (2003), 31–39.

[282] Oskar van Rest, Sungpack Hong, Jinha Kim, Xuming Meng, and Hassan Chafi. 2016. PGQL: a property graph query language. In *Proceedings of the Fourth International Workshop on Graph Data Management Experiences and Systems*. ACM, 7.

[283] Richard Van Heuven Van Staereling, Raja Appuswamy, David C van Moolenbroek, and Andrew S Tanenbaum. 2011. Efficient, modular metadata management with loris. In *Networking, Architecture and Storage (NAS), 2011 6th IEEE International Conference on*. IEEE, 278–287.

[284] M. A. Vef, R. Steiner, R. Salkhordeh, J. Steinkamp, F. Vennetier, J. F. Smigielski, and A. Brinkmann. 2020. DelveFS - An Event-Driven Semantic File System for Object Stores. In *2020 IEEE International Conference on Cluster Computing (CLUSTER)*. 35–46. https://doi.org/10.1109/CLUSTER49012.2020.00014

[285] Marc-Andre Vef, Rebecca Steiner, Reza Salkhordeh, Jorg Steinkamp, Florent Vennetier, Jean-Francois Smigielski, and Andre Brinkmann. 2020. DelveFS - An Event-Driven Semantic File System for Object Stores. In *2020 IEEE International Conference on Cluster Computing (CLUSTER)*. 35–46.

[286] Daniela Vianna, Varvara Kalokyri, Alexander Borgida, Thu D. Nguyen, and Amelie Marian. 2019. Searching heterogeneous personal digital traces. In *Proceedings of the Association for Information Science and Technology*, Vol. 56. 276–285.

[287] Daniela Vianna, Alicia-Michelle Yong, Chaolun Xia, Amelie Marian, and Thu Nguyen. 2014. A tool for personal data extraction. In *2014 IEEE 30th International Conference on Data Engineering Workshops*. 80–83.

[288] Kim J Vicente, Brian C Hayes, and Robert C Williges. 1987. Assaying and isolating individual differences in searching a hierarchical file system. *Human factors* 29, 3 (1987), 349–359.

[289] K. J. Vicente and R. C. Williges. 1988. Accommodating individual differences in searching a hierarchical file system. *International Journal of Human-computer Studies International Journal of Man-machine Studies* 29, 6 (1988), 647–668.

[290] Francesco Vitale. 2020. *Personal data curation in the cloud age: individual differences and design opportunities*. Ph. D. Dissertation. University of British Columbia.

[291] Francesco Vitale. 2020. *Personal data curation in the cloud age : individual differences and design opportunities*. Ph. D. Dissertation. University of British Columbia. https://doi.org/10.14288/1.0392427

[292] Francesco Vitale, Izabelle Janzen, and Joanna McGrenere. 2018. Hoarding and Minimalism: Tendencies in Digital Data Preservation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 587.

[293] Rachel Walton. 2015. Searching High and Low: Faceted Navigation as a Model for Online Archival Finding Aids (A Literature Review). *Journal for the Society of North Carolina Archivists* 12 (2015).

[294] Rachel Walton. 2017. Looking for Answers: A Usability Study of Online Finding Aid Navigation. *The American Archivist* 80, 1 (2017), 30–52.

[295] Cong Wang, Komal Thareja, Michael Stealey, Paul Ruth, and Ilya Baldin. 2019. COMET: A Distributed Metadata Service for Federated Cloud Infrastructures. In *2019 IEEE High Performance Extreme Computing Conference (HPEC)*. 1–7. https://doi.org/10.1109/HPEC.2019.8916536

[296] GuoRen Wang, HongJun Lu, Ge Yu, and Bin YuBao. 2003. Managing very large document collections using semantics. *Journal of Computer Science and Technology* 18, 3 (2003), 403–406.

[297] Richard Watson, Stijn Dekeyser, and Nehad Albadri. 2017. Exploring the design space of metadata-focused file management systems. In *Proceedings of the Australasian Computer Science Week Multiconference*. ACM, 20.

[298] Richard W Watson. 1981. Identifiers (naming) in distributed systems. In *Distributed Systems—Architecture and Implementation*. Springer, 191–210.

[299]]microsoft:refs:features Garrett Watumull, Jason Gerend, Liza Poggemeyer, Andrew Hansen, and Kaushik Ainapure. [n. d.]. Resilient File System (ReFS) overview. https://tinyurl.com/y832s3ky (accessed January 17, 2019).

[300] Jim Webber and Ian Robinson. 2018. *A programmatic introduction to neo4j*. Addison-Wesley Professional.

[301] Sage A. Weil, Scott A. Brandt, Ethan L. Miller, Darrell D. E. Long, and Carlos Maltzahn. 2006. Ceph: a scalable, high-performance distributed file system. In *Proceedings of the 7th symposium on Operating systems design and implementation*. 307–320.

[302] Kyu-Young Whang and Ravi Krishnamurthy. 1991. The Multilevel Grid File-A Dynamic Hierarchical Multidimensional File Structure.. In *DASFAA*, Vol. 1991. World Scientific, 449–459.

[303] Tmsu Wiki, Github Personal Open, and Business Explore. 2016. oniony / TMSU. (2016), 1–2.

[304] MV Wilkes. 1964. A programmer's utility filing system. *Comput. J.* 7, 3 (1964), 180–184.

[305] Lei Xu, Ziling Huang, Hong Jiang, Lei Tian, and David Swanson. 2014. VSFS: A searchable distributed file system. *Proceedings of PDSW 2014: 9th Parallel Data Storage Workshop - Held in Conjunction with SC 2014: The International Conference for High Performance Computing, Networking, Storage and Analysis* (2014), 25–30. https://doi.org/10.1109/PDSW.2014.10

[306] Zhichen Xu, Magnus Karlsson, Chunqiang Tang, and Christos Karamanolis. 2009. Semantic file system. US Patent 7,617,250.

[307] Zhichen Xu, Magnus Karlsson, Chunqiang Tang, and Christos T Karamanolis. 2003. Towards a Semantic-Aware File Store.. In *HotOS*. 181–187.

[308] So Yoon Yoon and Eric L. Mann. 2017. Exploring the Spatial Ability of Undergraduate Students: Association With Gender, STEM Majors, and Gifted Program Membership. *Gifted Child Quarterly* 61, 4 (2017), 313–327. https://doi.org/10.1177/0016986217722614 arXiv:https://doi.org/10.1177/0016986217722614