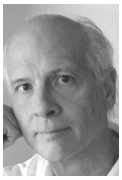
Jacob Farmer on Managing Research Data

An Interview

RIK FARROW



Rik is the editor of *;login:*.
rik@usenix.org

Jacob Farmer is an industry-recognized expert on storage networking and data protection technologies. He has authored numerous papers and is a regular speaker at major industry events such as Storage Networking World, VMworld, Interop, and the USENIX conferences. Inside the data storage industry, Jacob is best known for having authored best practices for designing and optimizing enterprise backup systems and for his expertise in the marketplace for emerging storage networking technologies. Follow him on Twitter @JacobAFarmer.
jfarmer@cambridgecomputer.com

I am always looking for interesting system administration articles, so when Doug Hughes told me that Jacob Farmer was working on a project where they tagged, that is, added metadata to files stored at the Harvard Medical School, I asked Farmer to write an article about their project. Prodding Farmer, in the form of emailing him questions, worked much better, as I discovered that even if I called in the evening, often Farmer would still be working on the project.

Rik: We heard through the grapevine that Cambridge Computer is doing some kind of data management project with Harvard Medical School that involves metadata, backups, and data migration. What can you tell us about the project?

Jacob: Indeed, my team is engaged with Harvard Medical School to do some novel things with regard to managing unstructured research data. The formal goal is to develop a more intimate understanding of how storage resources are consumed by various research groups. The folks at HMS feel that they are always reacting to storage technology demands. They feel that if they can better understand the needs of their researchers, they can be more proactive in providing storage resources. Like many data-intensive research organizations, HMS would like to have a long-term storage strategy, and that starts by sorting out what they have now and measuring the trends.

Rik: Is this kind of analysis a new service that Cambridge Computer is offering, or is this a one-off professional services engagement?

Jacob: My company has begun to offer this kind of analysis as a service, but what's really going on is that HMS has agreed to be an early adopter for a software product that my company is developing, and this project is simply the first of many projects that we hope to tackle with the software. About a year ago, I shared my product vision with the CIO at Harvard Med School. He really liked the vision and offered to be an early adopter/guinea pig. We have been running our software at HMS for about a year now across about 500 TB worth of files.

Rik: Does your software product have a name, and can you tell us in a nutshell what the vision is that HMS found so appealing?

Jacob: The code name for our product is Starfish. Starfish is a platform that enables users and applications to associate metadata with files and directories in conventional network file systems. The metadata is then used to enable better organization, more insightful reports, and to trigger storage management rules.

Our basic premise is that traditional file systems do not provide enough insight into the files they are storing to allow meaningful reporting or storage management policies. Meanwhile, more structured repositories and content management systems are too restrictive to be embraced by researchers. Researchers are notoriously resistant to efforts by the IT department to put structures and restrictions on the way they work. For instance, they resist directory naming conventions, seldom delete unneeded files, etc. Nonetheless, institutions need to impose rules and structure if they want to have a prayer at managing the explosive volumes of data typical of research computing. Our approach is to layer on the metadata and rules in an unobtrusive way.

Rik: That sounds like a very broad vision. What kinds of specific problems do you anticipate tackling with Starfish?

Jacob: I break the potential solutions into three top-level categories: IT infrastructure, curation/collaboration, and governance. When it comes to IT infrastructure, conventional solutions for backup, archiving, and tiered storage are all limited by the information available in file-system attributes. As such, their policies are not very granular, which is why conventional solutions fail when faced with hundreds of terabytes and zillions of files.

I then lump collaboration and curation together, because they both require meaningful metadata. Our software would enable collaboration by organizing files in ways that would be meaningful to third parties. As for curation, I don't anticipate that our metadata framework would meet the standards of a digital preservation professional, but we enable their mission by providing the lower-level storage management functions that their metadata systems do not perform, and we can facilitate the handoff from research to formal curation.

When it comes to governance, the combination of metadata and rules is very powerful. For instance, Starfish could be used to isolate files with data involving human subjects or export restrictions. Starfish could also be configured to provide administration or auditing for data management plans that were committed in a grant proposal.

Rik: Okay, that is a very broad vision. How does one product do all of that in any meaningful way?

Jacob: Admittedly, Starfish can't do all of these things, but we believe that all of these problem areas have some common elements, and Starfish will provide the framework for tackling a variety of hard problems with a single foundation. Most specifically, you can't do anything without having better insight into the business value of your files. To that end, Starfish provides a framework for associating metadata with files and directories. All of this information is incredibly valuable from a search and reporting standpoint, but it becomes even more valuable when you can do stuff based on the information.

Rik: Would you say that the project at HMS been a success so far? Do you have any data points you can share?

Jacob: I can certainly share a few highlights. We sampled roughly a half petabyte of miscellaneous files. We found that roughly 180 TB had not been touched in more than three years. There were roughly 20 TB of files with words such as *trash* and *junk* in the file name, and another 20 TB with *archive* in the file name. All told about 40% of the total file storage are candidates for being stored on a lower cost

tier. We still have to get user buy-in before moving any files, and we believe we have work to do to determine the most user-friendly way to migrate those files away and put them back if they are ever needed.

Rik: Just finding that out sounds useful to me. You mentioned that HMS has a laundry list of projects that they hope your software will enable. What's next?

Jacob: One of our next big projects is to help users visualize the cost of data storage, especially as they relate back to projects and grants. One possibility is to use the costing data for charge-backs. Another is simply to encourage good citizenship. Either way, try to imagine an individual storage consumer being able to visualize how much capacity and cost is in each folder in a directory and then having the tools to demote and promote files from one tier to another with full visibility into the cost implications. That's the kind of stuff we are hoping to do … and at very large scale.

Rik: I can imagine that a diverse research institute such as Harvard Med has at least one of every kind of storage management problem, so are you devoting 100% of your efforts on Harvard, or are you branching out with other early adopter clients?

Jacob: We have about a dozen installations, mostly in the life sciences, but we also have installations in financial services and in semiconductor manufacturing. In each case, the client has a specific objective, and we are working together to see how Starfish can help them meet the objective. For instance, at Fred Hutchinson Cancer Research Center, they are motivated by detailed reporting and by matching storage capacities to grants and projects. At Indiana University Bloomington, we are working together to define best practices for facilitating data management plans for grant-funded research. At Dana-Farber [Cancer Institute], we are exploring data protection and life-cycle management. Meanwhile, I have some commercial clients tinkering with using metadata to automate pipelines and others trying to use their automated pipelines as a way of capturing metadata.

Rik: You say that you are targeting research institutions. I would think these kinds of solutions would benefit any IT department in any industry. How are the needs of researchers different from those of traditional enterprise?

Jacob: Data-intensive research is a good niche for us. First, we have an extensive client base, so we are familiar with the problems, and people know us and are excited to work with us. Second, research data tends to flow through pipelines that really lend themselves to policy-based data management. Typically, there are raw data that fit the WORN paradigm: Write Once, Read Never. Then there are intermediate results, which often fit the WORSE paradigm: Write Once, Read Seldom if Ever. Then there are final results. Often there are multiple steps in the processes. Often there are offshoots of the main research.

Another interesting twist with researchers is that they are more concerned about preserving the old stuff than they are about protecting their most recently created files. New files often can be regenerated, but the old files might be needed to support a publication. For example, if researchers lose data from an experiment they ran yesterday, they can re-run the experiment. If they lose data from five years ago that formed the basis of the paper they are trying to publish, they might fail in their scientific mission. In a bank or insurance company, the business is much more concerned about minimizing downtime and preventing even the tiniest loss of newly created data.

Finally, researchers tend to have funky collaboration needs. Your typical enterprise only shares files within the enterprise, conforming to the paradigm of LDAP and POSIX permissions. Researchers often need to collaborate with other institutions or sometimes isolated users in their same institution. The granting agencies seem to be showing favor to programs that involve cross-institutional collaboration. Meanwhile, the granting agencies are mandating that researchers specify in their grant proposals what their plan is for retaining data and sharing data with interested third parties, which is a daunting problem when you have large data sets stored on a NAS behind the firewall.

Rik: Still, it sounds like this kind of technology could benefit other industries. Why limit yourself?

Jacob: Yes, there are potential applications for our software in other industries, but there are plenty of good companies pursuing intelligent file management for the traditional enterprise. Meanwhile, research is still a very broad niche. For instance, we have a lot of interest coming from libraries and museums. The libraries, museums, and other institutions of cultural preservation have very comprehensive metadata management systems for digital content curation, but they lack tools that interface with storage devices. For instance, digital librarians love our ability to verify the data integrity of their digital objects in an automated and audited way. They also like our ability to enable tiered storage and backups. One of the really cool things we are looking into is facilitating the data handoff between research computing and the libraries.

Rik: I would like to understand a bit more about the technology. For starters, I'm having a little trouble visualizing where Starfish sits relative to applications, users, and storage devices. It sounds like it would have to sit in the data path.

Jacob: Quite the contrary. Several vendors over the years have tried to virtualize file systems with devices that sit in the data path between the storage device and the users/applications. Just about all of these vendors died out early because these devices introduce latency, complexity, and often impose a least-common denominator effect on your fancy storage devices. Our model is to sit to the side of the storage device for I/O intensive workloads. We will sit in the data path for archival or cloud access.

Rik: If you are out-of-band, how do you control direct access to the files? Do you have to lock down the file servers somehow?

Jacob: You have hit on both our magic and our imperfection. Because we are not in the data path, we have to do our best to figure out what happened, where it happened, and when. Some NAS and file system products can produce a log or post events that we can monitor whenever there are changes made to the file system. It is also possible for us to learn of file system changes through the GUI or API. Worst case, we have to re-crawl the file system from time to time, but the good news is that we have a really fast crawler.

Rik: Okay, but what if a user deletes a bunch of files that are referenced by the metadata. How would you prevent that from happening?

Jacob: We can't really prevent a user from deleting files on a production file system, unless we programmatically modify permissions or apply a read-only flag. If there are metadata associated with deleted files, our system can report on the fact that the files are now gone. If you are really worried about files getting deleted, then you

configure our software to impose a backup policy that puts a copy of the file in a safe place. Now if the file is deleted, we can still associate the metadata with the backup copy and we can present an option to restore the file back to the production file server.

Rik: So, in summary, it sounds like you are crawling file systems on a regular basis and making a big database that tracks each file and directory. Is there any secret sauce or unique intellectual property that differentiates you in the marketplace?

Jacob: Candidly, our software is not doing anything that has not been done before by clever sysadmins or programmers around the world. What makes the software special is that we engineered enterprise-class software to do all of these things robustly and reliably and that scales to handle the capacities and numbers of files that you find in big research institutions.

I will give you an example from one of our early adopters. A few years ago, the client's IT department had written a Perl script to crawl their file systems, make a database record for each directory, and then associate directories with various engineering projects. The goal was to be able to look up a project in the database and see all relevant directories across all file servers and geographies. Similarly, they wanted to look up a directory and see what project it was associated with. They wrote the software. It worked. But then they grew from a handful of file servers to hundreds and from a few dozen terabytes to petabytes. The software buckled under the load. Then they had some turnover in the IT department, the code was abandoned, and now they have no solution to the problem. Our software just drops in and gives them the same functionality, just at a larger scale. We could do a whole lot more for them, but this is all they need and they can't seem to find it anywhere else other than writing it themselves.

Rik: When members of the USENIX community think of Cambridge Computer, we don't necessarily think of a software company. Does this project represent a departure from your traditional business model?

Jacob: Yes and no. Yes, in that this is the most ambitious software project we have ever taken on. We have done software applications before, but typically for very niche-y solutions or for developing tools for our field services people to gather metrics at our clients' sites.

That said, I feel that this project is otherwise a natural extension of my day job. In my traditional business, I work like a broker or agent. I help my clients narrow down and select the right storage technologies for their project. I try to do that without bias toward any particular vendors, and I get paid in the form of commissions when the client buys something. When my clients present me with a problem, I have all the incentive in the world to help them find the right vendor because that vendor will invariably pay me a finder's fee or commission. Over the past few years, my research clients have been presenting me with problems for which I can't find viable solutions, no matter how far and wide I go shopping. After a few years of this, I felt the inspiration to make it myself. In other words, I would never have the vision or the impetus for this project if I were not talking all day long to research institutions about their storage needs.

Rik: What kinds of resources are you investing in the project? Would you describe this as a skunkworks project, or something more formal?

Jacob: This is a real engineering effort with a seven-figure budget and full-time dedicated employees. My director of engineering is a former client of mine. In his past job, he built a SaaS application that provided e-discovery services for large legal cases. His system handled several billion files with all kinds of complex search and metadata. We are trying to build something similar, except even larger scale and for scientists instead of lawyers.

Rik: Have you considered making the software available as open source to encourage wider adoption?

Jacob: We have not made any decisions with regard to making the code available through an open source license. We are committed to openness by exposing a comprehensive API, but for now we feel we have to stay focused on building a really robust core product.

Rik: Final question: How can the USENIX community help?

Jacob: Two ways. For starters, we are hiring developers, so everyone please keep your ears open for developers who might be a fit for us. We are looking for professional software engineers with skills in large-scale, big databases, Python, Django, storage management, etc. We are also still open to taking on a few more early adopters, especially if they are willing to collaborate with us on defining feature sets and doing user-acceptance testing.