Master's Thesis Related Works

Michael Murphy

October 26, 2016

1 Related Works

1.1 Distributed storage and synchronization systems

1.1.1 Camlistore

Camlistore [4] is an open-source project to create a private long-term data storage system for personal users. It allows storage of diverse types of data and it synchronizes between multiple replicas of the data store. However, it eschews normal filesystems and creates its own schemas to store various media.

1.1.2 Dat Data

Dat [7] is an open-source project for publishing and sharing scientific data sets for research. This project has a lot of overlap with ours, and several of the core ideas are similar, including breaking files into smaller chunks, and tracking changes via a Git-like DAG. However, their focus is different. The Dat team is concentrating on publishing research data, and making that specific task as simple as possible for non-technical researchers who might not be familiar with version control. By contrast, our project operates at a lower level of abstraction, offering the full power of version control in a very general way, exposing and illuminating the complexities rather than trying to hide them or automate them away.

Where Dat focuses on publishing on the open internet, we focus on ad-hoc networks and data that may be private. Where Dat has components for automating peer discovery and consensus, we work at a lower level, trying to perfect and generalize the storage aspect first. Dat seems to assume that data sets will be small enough to fit on a typical disk on a workstation, while we want to scale even larger.

We hope that our system could be used as a base to build something like Dat, but we intend to create something even more general than the Dat core.

1.1.3 Eyo

Eyo [9] is system for storing personal media and synchronizing it between devices. It utilizes a Git-like content-addressed object database behind the scenes, but it works more like a networked filesystem than version control. It focuses on organizing media by metadata, which requires agreement on metadata formats, and it requires applications to be rewritten to access files via Eyo rather than the filesystem, both of which are thorny and ambitious problems. We prefer to focus purely on storage and synchronization.

1.1.4 git-annex and git-media

Git-annex [5] and git-media [2] are open-source projects that extend Git with special handling for larger files. Both store the metadata of larger files in the normal Git repository and then store the files themselves in a separate location. Git-media stores all the larger files in a separate data store which may be remote. Git-annex is more flexible. Annex files may be spread across several different remote repository clones or data stores, and git-annex has features for tracking the locations of annex files in different remote repositories and moving them from one repository to another. These tracking and distribution features are very similar to our goals. However, git-annex is not quite as flexible as we aim for in our system. It considers the large files atomic units, and it does not break them into smaller chunks for de-duplication. Also, because metadata is processed by Git, it has the same limitations that Git does. All repositories must have all metadata, and performance suffers when metadata is too large to fit into RAM.

1.1.5 IPFS: The Interplanetary Filesystem

IPFS [1] is an open-source project to create a global content-addressed filesystem. By its global nature, all files are stored together, publicly, in a global network of nodes with global addressing. IPFS should be an excellent resource for storing published information, but we wanted to work on a smaller, more private scale with discrete data sets. We want individuals and organizations to be able manage their own data stores privately on their own hardware.

It should be noted that IPFS does have support for storing private objects by way of object-level encryption. However, this seems wasteful of disk space, since small changes in the plain text of a file would completely change the ciphertext, leaving no way to compress the redundancy.

1.1.6 Kademlia

Kademlia [6] is an advanced distributed hash table system that updates its network topology information as part of normal lookups. It is an advanced piece of infrastructure, but like other distributed hash tables, it focuses on system-wide consistency, rather than the version-control paradigm we are trying to achieve.

1.2 Content-Addressed Storage and Backup

1.2.1 Boar

Boar [3] is an open-source project to create a version control system for large binary files. It is one of the main inspirations for our project. It stores file versions in a content-addressed way, and provides de-duplication for large files that only change in small pieces, and it can truncate history to reclaim disk space. However, Boar retreats to a centralized version control paradigm, with a central repository that working directories must connect to to check files in or out. We want to provide the advantages of Boar in a flexible distributed version control model. Boar also has practical limitations on repository size and number of files. Repositories are assumed to fit on one disk volume, and file metadata is assumed to fit into Ram. We aim to overcome both of those limitations.

1.2.2 Bup

Bup [8] is an open-source file backup system that is based on Git's repository format. A Bup backup is a valid Git repository and it can be read by Git, but Bup is a separate program written from scratch to read and write files to Git's pack file format directly, skipping Git's separate store and pack steps that use double the disk space. It has many features that we want for our low-level storage of the object database. It breaks files into chunks by rolling checksum, and it has considerations for metadata that is larger than RAM. However, it is locked into a backup-based workflow. History is linear and based on clock time of backup. And it assumes that the whole data set and the whole repository can fit onto one filesystem.

References

[1] Juan Benet et al. IPFS: The interplanetary filesystem. GitHub, 2014. `https://github.com/ipfs/ipfs`.

[2] Scott Chacon, Alex Lebedev, et al. git-media. `https://github.com/alebedev/git-media`.

[3] Mats Ekberg et al. Boar. `http://www.boarvcs.org/`.

[4] Brad Fitzpatrick et al. Camlistore is your personal storage system for life. `https://camlistore.org/`.

[5] Joey Hess et al. git-annex, 2015. `http://git-annex.branchable.com/`.

[6] Petar Maymounkov and David Mazières. *Kademlia: A Peer-to-Peer Information System Based on the XOR Metric*, pages 53–65. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[7] Max Ogden, Mathias Buus, Karissa McKelvey, et al. Dat Data. `http://dat-data.com/`.

[8] Avery Pennarun, Rob Browning, et al. bup, it backs things up. `https://bup.github.io/`.

[9] Jacob Strauss, Justin Mazzola Paluska, Chris Lesniewski-Laas, Bryan Ford, Robert Morris, and Frans Kaashoek. Eyo: Device-transparent personal storage. In *Proceedings of the 2011 USENIX Conference on USENIX Annual Technical Conference*, USENIXATC'11, pages 35–35, Berkeley, CA, USA, 2011. USENIX Association.