Syndicate: Democratizing Cloud Storage and Caching through Service Composition

Jude Nelson
Princeton University

Larry Peterson
Princeton University

The cloud is changing the way we share data. We can keep data on local workstations and file servers for quick access, but face the challenge of sharing it with a large number of people. Alternatively, we can put our data into one or more cloud storage systems to share it with many other users, but then we cannot access it as quickly or as cheaply. Moreover, local copies of data can get out of sync with cloud copies, causing remote users to see old versions. Our solution is Syndicate, a virtual cloud storage system that composes local storage, cloud storage, and commodity CDNs and network caches to transparently give users the best of both worlds.

Syndicate organizes a collection of data spread across multiple clouds and end-user computers into a *Volume*. Using Syndicate, a user sees and interacts with all data in a Volume as if it were a set of files and directories on local storage. All the while, Syndicate lets a scalable number of users read and write the data, ensures that they see a consistent view of the data, and keeps the data durable by uploading it to existing cloud storage providers.

To use Syndicate, a user installs a *Syndicate Gateway* (SG) on their local workstation or cluster. The SG presents all the files in a Volume (Figure 1), while serving locally-written data to a scalable number of SGs via existing, unmodified network caches and CDNs. They coordinate via a scalable *Metadata Service* (MS) to help them discover data, mask failures, and read fresh data from caches, even if some caches serve stale data.

Additional SGs can be configured to archive a Volume's data to one or more cloud storage services; Amazon S3 and Glacier [1], DropBox [4], Box.net [2], and Google Drive [6] are currently supported. Whenever a user writes data, cloud-facing SGs receive and upload it to their corresponding cloud storage providers. Then, if later one or more of the user-facing SGs fails, Syndicate is still able to read the data from one of these cloud-facing SGs.

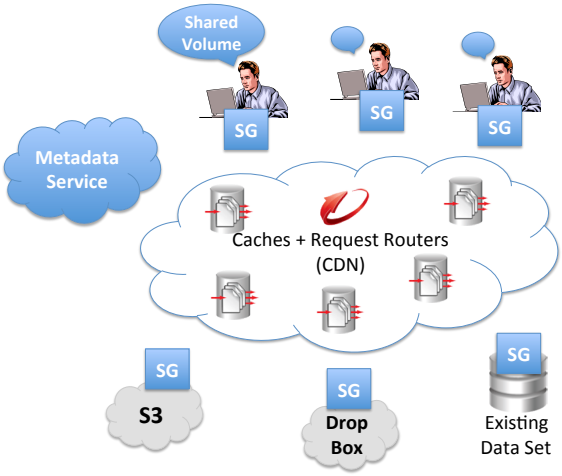

Figure 1: *Syndicate components run on top of existing network caches and cloud storage systems, making far-flung data appear as a locally-mounted Volume.*

Sometimes users need to work with an existing, remotely-hosted dataset, such as GenBank [5], CommonCrawl [3], M-Lab measurements [7], and so on. To allow this, an SG may also be configured to expose an existing dataset within a Volume as a read-only directory hierarchy. Syndicate downloads dataset records on-demand via these SGs, thereby transparently leveraging existing caches to scale read delivery.

By deploying the appropriate SGs, developers can create their own virtual storage systems layered "on top" of existing providers, letting them meet cost, performance, consistency, and durability requirements independent of the underlying implementations.

SoCC'13, 1–3 Oct. 2013, Santa Clara, California, USA.
ACM 978-1-4503-2428-1. http://dx.doi.org/**10.1145/2523616.2525954**

References

[1] Overview of Amazon Web Services. https://d36cz9buwru1tt.cloudfront.net/AWS_Overview.pdf.

[2] Box.net. http://www.box.net/.

[3] Common Crawl. https://drive.google.com/.

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

[5] GenBank. http://www.ncbi.nlm.nih.gov/genbank/.

[6] Google Drive. http://www.measurementlab.net/.

[7] Measurement Lab. http://www.measurementlab.net/.