Give Your Data the Edge: A Scalable Data Delivery Platform

Arizona, UNC, Open Networking Lab, Princeton

Science is increasingly data-driven. Collaborators regularly tap data from multiple repositories across the world to generate even more data for others to consume. Effective science requires effective data management, but most scientists are not experts in distributed storage. To address this, we created Syndicate, a general-purpose scalable storage system that automates data retrieval, staging, and storage across multiple sites.

Syndicate offers a sustainable way for curating scientific data with an emphasis on ease-of-use. It interfaces with legacy data storage systems, commodity cloud storage, and content distribution networks (CDNs) to create coherent read/write storage volumes that span multiple sites and attach to scientists' VMs and laptops like removable drives. Using only her workstation, a PI can select datasets and storage infrastructure, create volumes, and add collaborator user accounts, and the system will automatically configure itself at runtime.

The most significant challenge in the design and implementation of Syndicate is to provide end-to-end data gurarantees like consistency, durability, and availability. End-to-end consistency is particularly tricky because data analysis tools assume very particular consistency models, but the underlying infrastructure offers variable and often very weak guarantees. For example, BLAST interacts with data under POSIX consistency semantics, but an Amazon S3 bucket augmented with a CDN can at best provide delta consistency.

Mismatches like this result in lots of "glue code" to make scientific software interoperate with storage infrastructure. Glue code is workflow- and storage-specific, and maintaining it requires deep knowledge of both. This imposes high operational costs, since any change means writing more glue code to preserve compatibility.

Syndicate overcomes this challenge with a novel storage programming model that replaces bespoke glue code with *composable, reusable storage drivers* running in a *programmable storage fabric* that spans the wide-area (Figure 1). Once written, a Syndicate driver can be deployed and combined with other drivers and used in other workflows to provide the desired end-to-end consistency.

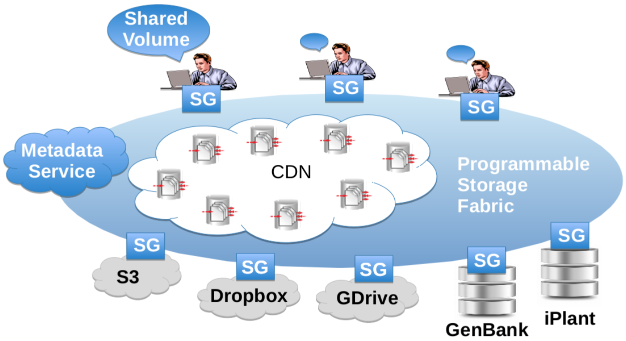

Figure 1: *Syndicate deployment. Drivers run in gateways (SG), which coordinate via a Metadata Service. Together, they make up the programmable storage fabric that links existing infrastructure together to create shared volumes.*

To achieve this, Syndicate provides built-in snapshot isolation on files "on top" of the storage infrastructure. Data chunks are *immutable* between writes, and have *globally-unique addresses* in the system. Whole-chunk `gets`, `puts`, and `deletes` never conflict, which lets users implement consistency simply by constraining the order in which chunks are processed relative to other chunks. By separating data storage and retrieval from operation ordering, we enable consistency-preserving logic to be developed independently of both the workflow and storage provider.

We have deployed Syndicate on OpenCloud to give each user a shared volume for accessing data across compute clusters. Users attach iRODS deployments to an Akamai CDN and commodity storage to accelerate data delivery across the wide-area and automatically stage it to compute clusters. The CDN's weak consistency does *not* affect workflow correctness. We have achieved this with only a few hundred lines of Python in the drivers.