Sirocco: A Storage System Design for Exascale

Matthew L. Curry[1], Geoff Danielson[2], H. Lee Ward[3], Anthony Skjellum[4], and Jay Lofstead[5]

Many existing parallel file systems offer storage for very large files with high performance by striping data across devices. Each of these systems have been optimized in different ways, but are at their core inspired by the Zebra file system [1], which also statically stripes data across servers. For exascale systems, the explosion of devices (in number and type) presents a challenge to the inherently flat architecture of striped organizations.

Another challenge not addressed by current parallel file systems is the increasingly complex requirements for new applications. While current POSIX-centric file systems for HPC are geared toward traditional scientific computing workloads, data analytics applications are becoming a more prominent consumer of compute cycles on large-scale computers. Such applications need greater support from the storage system to enable richer, more capable, and more fault tolerant treatment of data at a finer grain.

Sirocco is a massively parallel, high performance storage system for the exascale era that breaks from the classical Zebra-style file system design paradigm. Its architecture is inspired by peer-to-peer and victim-cache architectures, and emphasizes client-to-client coordination, low server-side coupling, and free data movement to improve resilience and performance. By leveraging these ideas, Sirocco natively supports several media types, including RAM, flash, disk, and archival storage, with automatic migration between levels. Figure 1 shows how data can be written into Sirocco, and how Sirocco will automatically move data to satisfy safety constraints and enable higher performance.

Sirocco's design ensures scalability by its minimalist design. Sirocco is not a file system – Instead, it is a distributed object system that can be used as a storage component for a file system. This design is adherent to the Lightweight File System's [2] philosophy, which uses a minimal storage system with authentication and authorization. Other services, such as naming and locking, are separate components. While system-level versions of these services exist, clients that implement the file system interface opt in to using them, and may choose the right kind for their workload.

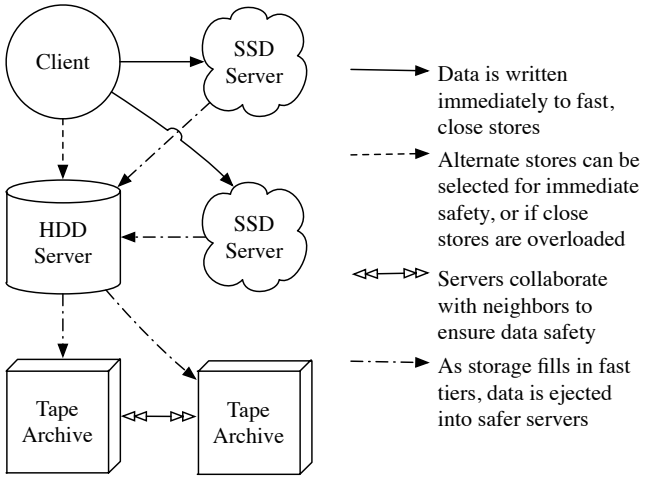

Fig. 1: Data moves from clients through Sirocco based on load, capacity, and desired safety. Note that these behaviors are based on local decisions; there are no explicit tiers, but tiered behavior follows from safety-motivated victim caching.

Another differentiating feature of Sirocco is that it includes storage interfaces and support that are more advanced than typical storage systems. For example, Sirocco enables clients to efficiently use the key-value or bulk storage paradigm with the same interface. It also provides several levels of transactional data updates within a single storage command, up to and including ACID-compliant updates. This transaction support extends to updating several objects within a single transaction. Further support is provided for optimistic concurrency control, enabling greater performance for workloads while providing safe concurrent modification. These facilities enable efficient file system client implementation, while also providing building blocks for tools to work with advanced data models.

For news of further Sirocco availability and other artifacts, see `http://www.cs.sandia.gov/Scalable_IO/sirocco`.

[1]Sandia National Laboratories, Center for Computing Research. mlcurry@sandia.gov

[2]Hewlett Packard Enterprise. geoffrey.danielson@hpe.com

[3]Sandia National Laboratories, Center for Computing Research. lee@sandia.gov

[4]Department of Computer Science and Software Engineering, Auburn University. skjellum@auburn.edu

[5]Sandia National Laboratories, Center for Computing Research. gflofst@sandia.gov

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energys National Nuclear Security Administration under contract DE-AC04-94AL85000.

REFERENCES

[1] J. H. Hartman and J. K. Ousterhout, "The Zebra striped network file system," *ACM Trans. Comput. Syst.*, vol. 13, no. 3, pp. 274–310, Aug. 1995. [Online]. Available: http://doi.acm.org/10.1145/210126.210131

[2] R. Oldfield, L. Ward, R. Riesen, A. Maccabe, P. Widener, and T. Kordenbrock, "Lightweight I/O for scientific applications," in *Cluster Computing, 2006 IEEE International Conference on*, Sept 2006, pp. 1–11.