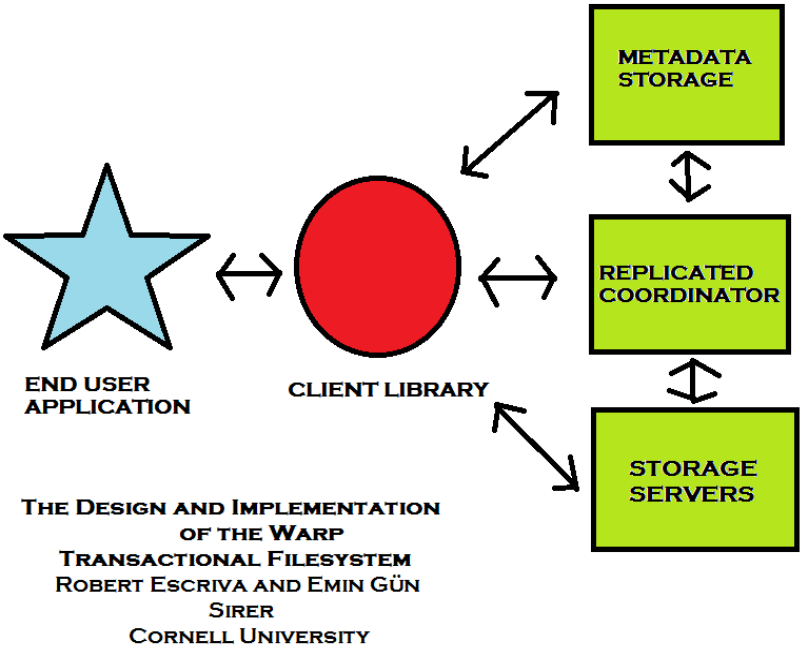
Name : Muhammad Ishraf Shafiq Zainuddin
ID : 200342741
Assignment : 3 (Research Paper)

The Design and Implementation of the Warp Transactional Filesystem

by : Robert Escriva, Emin Gun Sirer
Computer Science Department, Cornell University

The current filesystems exhibits a tension between achieving a high performance for the distributed setting or retaining the familiar semantics of local filesystems (Escriva, Sirer). Filesystems design not only require special hardwares which can be cost prohibitive but also jeopardizing consistency for performance or artificially restrict the filesystem interface (Escriva, Sirer). This paper introduces Warp Transactional Filesystem (WTF), a POSIX-compatible filesytem which enables new class of high performance applications to efficiently read, write and rearrange files without rewriting the underlying data based on a new file slicing operations API (Escriva, Sirer). With this, not only garbage collection and database can be compressed without writing the data, but multiple files can be linked together without them being read, and even contents of record-oriented files can be sorted without rewriting the files' contents (Escriva, Sirer).



WTF's distributed architecture consists of four components which are the (1) metadata storage that is builds on top of HyperDex and its expansive API, (2) the storage servers which contains filesystem data and provisioned for high I/O workloads, (3) the replicated coordinator which is a rendezvous point for all components of the system while maintaining list of storage servers and last but not least (4) the client library that contains the majority of the functionality of the system and is where WTF combines the metadata and data into a coherent filesystem (Escriva, Sirer). File is represents as a sequence of byte arrays by WTF that, when overlaid, comprise the file's contents. The central abstraction is a slice, byte-addressable, arbitrarily sized sequence of bytes while a file in WTF, then is a sequence of slices and their associated offsets (Escriva, Sirer). The abstraction provides a separation between metadata and data that enables

filesystem-level transaction to be implemented using, solely, transaction over the data (Escriva, Sirer). Data is stored in the slices, while the metadata is a sequence of slices. WTF can transactionally change these sequences to change the files they represent without rewriting the data (Escriva, Sirer).

File slicing abstraction greatly simplified the design of storage servers which deal with slices and are oblivious to files, offsets, or concurrent writes (Escriva, Sirer). The minimal API required by file slicing consists of just two calls which are to create and retrieve slices. Furthermore, a storage server processes a request to create a slice by writing the data to disk and returning a slice pointer to the caller (Escriva, Sirer). WTF partitions a file into fixed size regions with each of its own list. Each regions is stored as its own object in HyperDex under a deterministically derived key to achieve support for both arbitrarily large files and efficient operations on the list of slice pointers (Escriva, Sirer). It would be impractical to keep the list of slice pointers small by limiting the number of writes to a file even though practically, it is desirable so that they can be stored, retrieved, and transmitted with low overhead (Escriva, Sirer). Last but not least, due to the file slicing interface, applications able to manipulate subsequences of files at the structural level without copying or reading the data itself instead of operating on bytes as traditional POSIX systems do (Escriva, Sirer).

The author concluded that WTF achieves throughout and latency similar to industry-standard HDFS in a broad evaluation, while providing a stronger guarantees and richer API simultaneously (Escriva, Sirer). WTF which is a new distributed filesystem that enables applications to operate on multiple files transactionally without requiring complex applications logic together with the new filesystem abstraction file slicing that changing the filesystem interface to focus on metadata manipulation instead of data manipulation are a potent combination that enables a new class of high performance applications (Escriva, Sirer).

The conclusion does support the thesis since experiments show that WTF can qualitatively outperform the industry-standard HDFS distributed filesystem, up to a factor of four in a sorting benchmark, by reducing I/O costs (Escriva, Sirer). This is also supported in Luis-Felipe Cabrera and Darrell D. E. Long. Swift: Using Distributed Disk Striping to provide High I/O Data Rates in Computing Systems (1991).

References

Escriva, R. and Sirer, E. (2016). *The Design and Implementation of the Warp Transactional Filesystem*. [online] Usenix.org. Available at: https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-escriva.pdf [Accessed 11 Jun. 2018].

Luis-Felipe Cabrera and Darrell D. E. Long. Swift: Using Distributed Disk Striping To Provide High I/O Data Rates. In Computing Systems, 4(4):405- 436, 1991.