Michael Berlin

Consistent and Partition-Tolerant File Replication in the Cloud File System Xtreem FS

Michael Berlin: E-Mail: berlin@zib.de

I XtreemFS – A Cloud File System

While cloud providers use shared distributed hardware, which is inherently unreliable and insecure, cloud users expect their data to be safely and securely stored, available at any time and accessible in the same way as their locally stored data. As a solution we present XtreemFS, a file system for building reliable cloud infrastructures on distributed off-the-shelve hardware.

XtreemFS is a distributed, POSIX compatible file system that provides advanced features like transparent replication and volume snapshots. It is based on the scalable object-based storage architecture [1] and thus allows to dynamically add and remove storage capacity. It supports SSL connections and X.509 certificates and is therefore suitable for deployments in wide area networks.

II XtreemFS File Replication

While the main goal of replication in a distributed storage system is to ensure availability, other issues like the consistency of the replicas and the system's ability to cope with the partitioning of the network also have to be considered. These system properties are stated as consistency (C), availability (A) and partition-tolerance (P). According to Brewer's CAP theorem [2] at most two out of the three properties can be achieved, and therefore storage systems support a different set of properties depending on the application. For example, most cloud storage services like Amazon S3 are available, partition-tolerant (AP) systems. On the contrary, high availability solutions for clusters like DRBD support consistency and availability (CA) and therefore are prone to so-called 'split-brain' situations.

In practice, consistency cannot be forfeit if the interface (e.g., the POSIX file system interface) or the application is not able to report and resolve conflicts. Although network partitionings happen seldom, they have to be considered. Therefore, the XtreemFS file replication protocol is designed as a CP system which guarantees consistency and tolerates network partitionings. The system is quorum based and also provides availability as long as a majority of replicas can be reached.

Internally, the protocol employs the primary/backup scheme and uses leases to enable primary failovers. A lease is coordinated among the file's replicas without a central component by the Flease algorithm [3], which is based on Paxos [4]. Partition-tolerance and consistency are achieved by always operating on a majority of replicas and synchronously updating replicas. Additionally, our protocol out stands other solutions in the following regards:

a) Scalability: The file replication works at file granularity and scales with the number of files and servers as it involves no central components for issuing leases.

b) Efficiency: Unlike other Paxos-based solutions, the Flease algorithm requires no writes to persistent storage since it exploits the fact that leases can be discarded after their expiration. Thus, the file replication puts no additional load on the I/O subsystem.

Acknowledgement

We thank Björn Kolbeck for his work on the Flease algorithm and the file replication protocol. Financial support came from the EU projects XtreemOS (2006–2010) and Contrail (2010–2013, grant agreement FP7-ICT–257438) and also from the German BMBF project MoSGrid (2009–2012).

References

1 M. Mesnier, G. Ganger, and E. Riedel, "Object-based storage," *IEEE Communications Magazine*, vol. 8, pp. 84–90, 2003.

2 E. Brewer, "Towards robust distributed systems (abstract)," in *Proceedings of the Annual ACM Symposium on Principles of Distributed Computing*.

3 B. Kolbeck, M. Högqvist, J. Stender, and F. Hupfeld, "Flease – lease coordination without a lock server," in *25th IEEE International Symposiumon Parallel and Distributed Processing (IPDPS2011)*, 2011, pp. 978–988.

4 L. Lamport, "The part-time parliament," *ACM Transactionson Computer Systems*, vol. 16, no. 2, pp. 133–169, 1998.