BookKeeper and GlusterFS

A comparison between BookKeeper and GlusterFS.

GlusterFS also implements striping and replication (GlusterFS). However, GlusterFS has been designed to be a general purpose file system and it does not provide the same guarantees that BookKeeper for journalling/WAL as we discuss next.We first discuss our understanding of the GlusterFS architecture and point to the differences we see.

Overview of GlusterFS

General setup

In a distributed GlusterFS setup, there is a daemon, glusterd running on all the machines. The administrator uses gluster to create a /trusted server pool/ from these machines. All machines in the pool then share the configuration. The pool is configured by the commandline tool.

Servers in the pool expose bricks[1], which represent the lowest level of storage in the system, which is basically a filesystem partition. The administrator can then compose and layer /translators/ on top of this to build a volume. Translators are conceptual objects which expose a filesystem interface. For example, the AFR translator[2] provides replication by writing all changes to a number of child translators. At the lowest level there is a Posix translator which sites directly on top of a brick.

AFR replicates by write all changes to all child translators, much like RAID1. If a child is down, write will go to the other children. If all children go offline, the volume itself goes offline.

Consistency issues.

Imagine a system with a AFR translator replicating to two Posix translators (A & B). If one replica(A) goes offline, a client will continue to write to the other (B). Now, B goes offline. The client will stop writing to any replica. A is restarted so the client can continue writing. The client writes updates to A. Now A and B are inconsistent.

Differences to BookKeeper

[1] http://www.gluster.com/community/documentation/index.php/GlusterFS_Concepts
[2] http://www.gluster.com/community/documentation/index.php/Understanding_AFR_Translator