Isaac Semaya & Michael Xu .
CSCI1380 — Spring 2018

Puddlestore

May 27, 2018

1 Introduction

Puddlestore is a distributed file system built off of two previous CSCI1380 projects: Tapestry and Raft.

2 Background

2.1 Tapestry

Tapestry is a distributed hash table (DHT). Nodes can publish key-value pairs. Lookups can be done from any node by providing the appropriate key. Nodes are able to join and leave Tapestry without compromising the integrity of the data[1].

2.2 Raft

Raft is a consensus algorithm that allows multiple nodes in a cluster to maintain identical state machines. All requests to these state machines are totally ordered with respect to the time at which the leader node receives the request. Raft is fault tolerant such that unreliable nodes (nodes that disconnect from the network, run slowly, etc.) do not affect the overall integrity of the state machine[2].

3 Abstract

Within Puddlestore, all files and directories are stored as blocks in Tapestry. Each block has attributes, one of which is a globally unique identifier (GUID), which serves as its key in Tapestry. Some blocks may store metadata about files, others may store raw data, and some may store GUID references to other blocks. Given that updates to blocks need to be totally ordered, blocks need to be updated or deleted in such a way that preserves atomicity. Since Raft provides a framework that allows multiple clients to maintain identical state machines (as mentioned in **2.2**), if given a state machine that maps GUIDs to GUIDs, Raft will be able to update blocks while maintaining the properties mentioned above. Puddlestore uses Raft's safety and atomic properties to perform Copy On Write modifications. More specifically, every block has an AGUID (Active GUID) and a VGUID (Version GUID). The AGUID is constant for the life of the block and is used by other blocks to reference it while the VGUID represents a specific copy of that block and is only known to the block itself and to Raft. When a change is made to a block, that block is copied within Tapestry (and thus given a new VGUID), that copy is modified and the AGUID to VGUID mapping is updated in Raft. Thus, when changes are made to a block, its update is only apparent to blocks referencing it when the AGUID to VGUID mapping is updated. Thus, Puddlestore ties Tapestry and Raft together with a Bash-like shell interface that allows for the safe creation, deletion, and modification of files and folders from an arbitrary number of clients.

[1] Given that the number of nodes leaving is not greater than the replication number specified
[2] Given that the number of unreliable nodes is less than half of the total nodes in the cluster

Modifications were also done to Tapestry to optimize both its performance and reliability. Within Tapestry, modifications were made such that nodes or paths which requested the same key multiple times would cache the value or location of those values locally. Reliability was improved with replication. Replication ensured that a certain number of copies of a KV pair existed on Tapestry at one time. If nodes crashed holding replicated values, those values would be rereplicated.