Mid-Term Report : Erasure Coding and Replication for Edge Devices

Sheshadri K R

Department of Computational and Data Sciences
Indian Institute of Science, Bangalore 560012 INDIA
Email: sheshadrik@iisc.ac.in

1. Objective and Goals

Given a file , its size and a storage budget.

The objective is to build a reliable storage at the edge layer to store this file, by using the collective storage space of all the edge devices. The objective is achieved using a hybrid technique which involves erasure coding (Reed-Soloman 6,4 erasure coding) and replication. I call the technique as Partial Replication Partial Erasure Coding (PRPE).

2. Problem Description

A storage budget is a decimal value which ranges between 1.5 to 3. 1.5 being minimum and 3 being maximum, this is specified by the user. The reason the minimum value is 1.5 is that, given a block of size 1 unit, Reed-Soloman 6,4 encoding takes 1.5 times the size of the orignal block to give a fault tolerance of up to 2 nodes going down.

Similarly, the reason maximum value is 3 is that, a triple replication takes 3 times the size of the orignal block to give a fault tolerance of up tp 2 nodes going down.

3. System Model

In the system model, we have 2 types of resources, first is Fog, the second is Edge. Fog is an equivalent of a desktop computer which has configuration of Intel(R) i5-8250U CPU @ 1.60GHz, 8GB of installed RAM , 500GB of storage. Edge devices are 4-core AMD CPUs with, 1GB of installed RAM and about 64GB of storage.

Design goals and assumptions:

- The clients which does the "put" and "get" are on the Edge devices.
- The edges are the primary storage devices.
- Irrespective of the technique being used to store the data, it should support resillience to failure upto 2 devices.
- The proposed PRPE technique should scale with joining and leaving of Edge devices.

The Fog and edge devices are in a common LAN.

3.1. Metadata management

The masternode manages the metadata of all the files that are stored in the datastore. The metadata related to a file includes the following information

- A file to block mapping
- A block to set of edge location mapping
- An edge to set of blocks mapping
- A block to type of encoding mapping (To differentiate between replication and Erasure coding)

I use the following equation to decide on how many blocks are to replicated and how many are to be erasure coded.

$$Nr = (2/3) * SB * totalBlocks \tag{1}$$

$$Ne = totalBlocks - Nr \tag{2}$$

Where,

- $Nr \rightarrow$ number of blocks that can be replicated
- $Ne \rightarrow$ number of blocks that needs to be replicated
- $SB \rightarrow$ storage budget given by the user
- $totalblocks \rightarrow$ total number of blocks for a given file.

Using this we can calculate the number of blocks that need to be stored using erasure coding, as shown in eqn 2.

3.2. Hardware and Software Requirements

To conduct these evaluations we need a set of 10 containers each to run the edge devices server, and a single node to run the master node.

The hardware requirements for the master node is. (this may change later, in such circumstances it'll be duly reported)

System Environment of Master Node:

- **CPU :** Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz
- **Total Installed RAM:** 8GB
- **CPU Architecture :** 64-bit
- **Number of Processing Cores :** 8
- **Java Version:** 1.8.0201
- **Message Exchange format:** Thrift Messages
- **OS :** Ubuntu 18.04

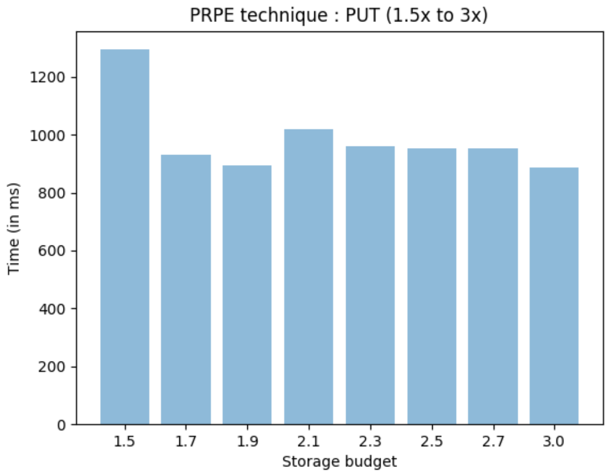

Figure 1: 50MB put file

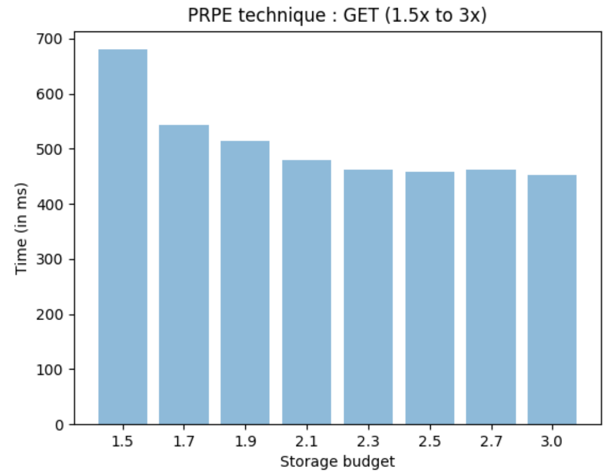

Figure 2: 50MB get file

The edge servers were also ran on the same machine where the master server was running.

4. Experimental Design

10 instances of edges are spawned in the same computer environment where the Master server is running.

4.1. Dataset for experiment

Used a file of size 1MB (1.08MB to be precise).

The experiments that I wish to carry are as stated below:

- Measure the read/write latency of putting a file in the data store with full replication/full erasure coding.
- Measure the read/write latency of putting a file in the data store with repliation-erasure coding method conduct the experiments for the following set of values 1.7x, 1.8x,1.9x, 2.0x, 2.1x, 2.2x, 2.3x , 2.4x, 2.5x
- Plot the graph of how the read/write latency goes from 1.5x to 3x.

4.2. Put Experiment

In this I put 50 different files of size 1MB (1.08MB) to the datastore. I varied the storage budget from 1.5X to 3.0x. For 1.5X all the files were fully erasure coded. At 3.0X all the files were fully replicated.

The time taken to write the blocks are shown in the plot 1.

Replication was expected to perform faster puts since it doesn't involve any calculations.

4.3. Get Experiment

In this, for all the 50 files I put,I did a get on them. Again files which were stored with different storage budget

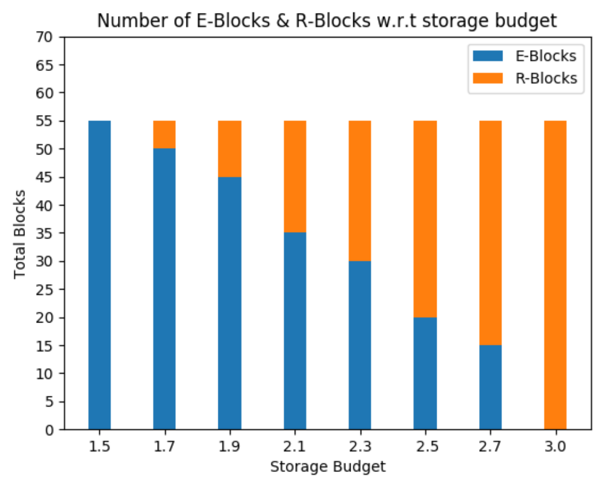

Figure 3: E-Blocks vs R-Blocks

were read. A get request of a file which is fully erasure coded was expected to be slow, while the file which was stored with fully replicated was expected to be fast.

The time taken to write the blocks are shown in the plot 2.

We should read the results of put and get experiments with the change in the number of replicated blocks (R-Blocks) and Erasure Coded blocks (E-blocks). It can be seen that, when we have more and more replicated blocks, the put and get experiments will perform well.

The number of R-Blocks and E-Blocks as we vary the storage budget is shown in the ploy 3

5. Mid Term deliverables

- Build a simple client-server system to support Partial-Replication Partial-Erasure Coding(PRPE).

- Given a file and a storage budget, store it using the PRPE technique and compare the read/write performance against fully replicated system to fully erasure coded system.
- Compare the recovery times for single failure for PRPE, with fully replicated and fully erasure coded.

References