2022-2023

Controlled Redundancy P2P Distributed Data Storage

Under the guidance of **Dr. Pankaj**

By **Rohan Sharma** & **Rishabh Singh**

Table Of Contents

Overview

A distributed data storage system would make data almost invulnerable to being lost or deleted as is possible in the current centralized model of the Internet. The current solution to this is torrenting. While popular and hugely successful, torrenting has a major disadvantage; the users willing to participate in it need to download and seed a huge amount of data.

We propose a possible solution with a network built on top of existing torrenting protocols. The network tracks the redundancy rate of each file in a network and consequently instructs the nodes to download files that are below the redundancy target value. It can also inform the nodes which files are heavily seeded and thus allow them to delete the files and reclaim disk space.

Problems to solve

1. Maintaining the target redundancy rate for each file throughout the network regardless of file size and peer count

2. Compensating for a low redundancy rate by starting file seeding throughout the network by using regular polling

3. Decreasing network load on individual peers and reducing memory footprint on every node that partakes in the network

4. Combating byzantine nodes and having multiple layers of security in beacon nodes.

Project objective

The project aims to solve the problem of participating in a P2P network serving large sized files. Our implementation aims to replicate only some files to some clients that will ensure that no one peer has to burden itself with downloading and seeding and the high memory requirement that comes with it.

It also aims to create a fault tolerant network wherein a number of peers/nodes going down would not affect the working nodes and the network would adjust to the current network condition.

Understanding the need for controlled redundancy

Existing Implementations

01

In this section we will be taking a look at existing technologies and comparing them to our own.

Current Implementations:
The most popular technology currently being used for P2P file transfer is Bittorrent. It allows for seeding and downloading files without the need for a centralized server. This is very effective however it suffers from the aforementioned problems of high bandwidth and memory requirements.

Existing Implementations

02

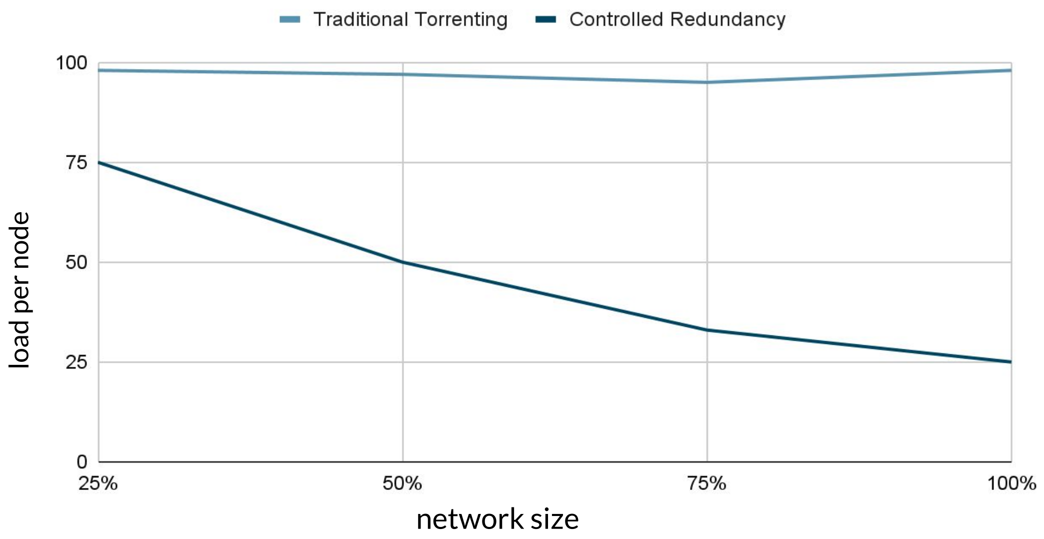
There also exist some redundancy controlled storage solutions but they either lack P2P or have different use cases, like databases.

Cassandra DB

It is a distributed, wide-column store, NoSQL database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. Cassandra offers support for clusters spanning multiple datacenters, with asynchronous masterless replication allowing low latency operations for all clients.

Load analysis

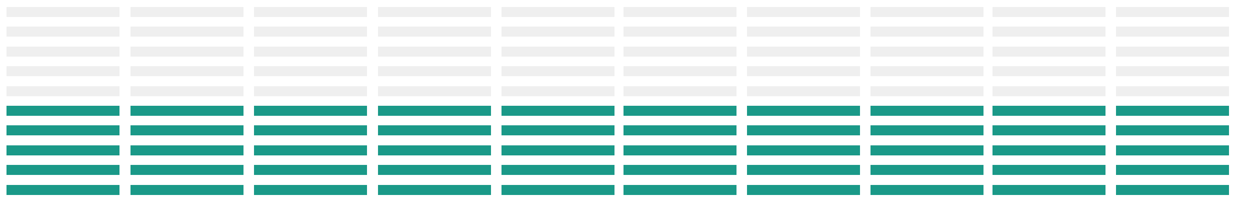
A good way of understanding the need for such a network would be to look at the difference in loads on peers which using the traditional torrent technologies and a controlled redundancy network.



The reducing of network load with increasing peers means that as a network grows, more peers can take part in it which reduces the load even further. This incentivizes more people to join and thus forms a feedback loop.

Since the load on every peer is lowered, lesser cost is required for setting up the necessary network and storage hardware.

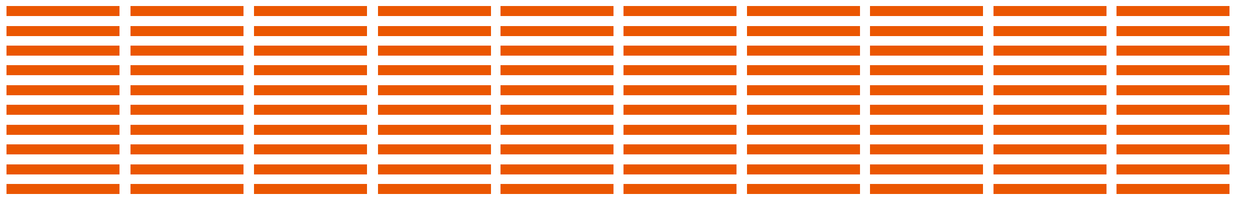
Controlled Redundancy



<40%

All peers share the load equally and thus it is reduced as the network grows

Traditional Torrenting



>90%

The peers do not share load and it remains high on all peers regardless of network condition.

Target audience

This project aims to develop a product that would target the following people

01 | Large enterprises that need to store public data

02 | Regular users who want to contribute to securing data

03 | Large scale databases like Wikipedia/Z-Library

04 | People who want to circumvent national censorship

05 | Data Hoarders

Proposed Methodology

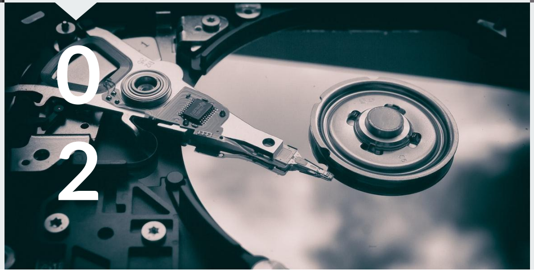
The new network will run on the existing torrent network owing to its widespread popularity and proven effectiveness.

However it will implement an addition layer on top of bittorrent which will allow for the tracking of redundancy rate of individual files and controlling them.

Process

Step 1

The initial peer creates and uploads a table of magnet links to a beacon node



Step 2

Contributing nodes query the beacon node for the link table and the redundancy rates for each file

Step 3

The nodes selectively download the files that do not match redundancy target

Technologies

This project will make use of the following modules and technologies:

1. NodeJS
2. BitTorrent
3. SQL
4. Redis
5. ReactJS

Backend

All peers will use NodeJS on the backend

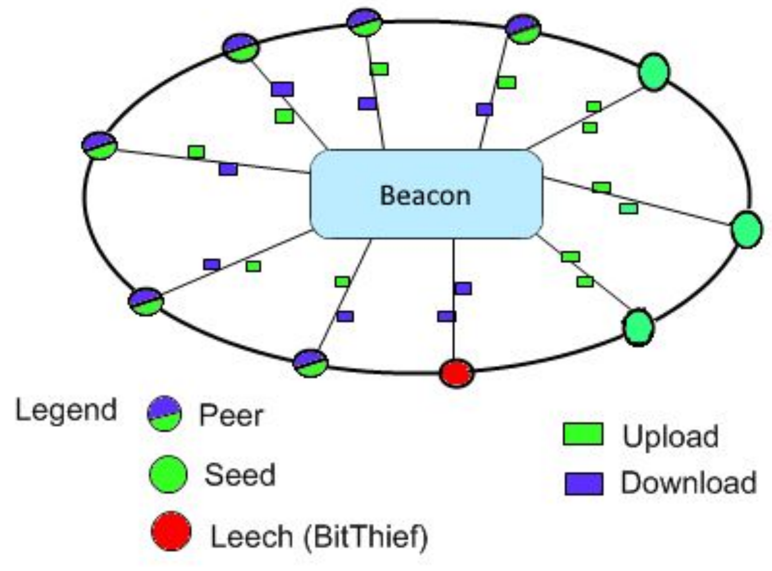
Frontend

ReactJS will be used for frontend rendering

P2P

The entire network will use bittorrent as a backbone.

Block Diagram



Real World Impact

The real world impacts of this project are

01 | Easier access to worldwide distributed data

02 | Participation in large networks is possible on handheld devices

03 | Data integrity is preserved

04 | More people are encouraged to take part in the network

05 | Network is flexible enough to sustain a heavy blow to mission critical nodes

Thank you.