

FINAL PROJECT PROPOSAL

Members

- Anying Li (anyingli)
- Lara Timbo Araujo (larat)
- Vinay Mayar (vmayar)

Problem

A large percentage of content loaded by websites is static content--images, videos, scripts, etc. It's expensive for most websites to distribute this content to clients directly. Many websites use CDNs to distribute static content. A cheaper alternative is to allow clients to load static content from each other. We wish to do this without placing any burden on the client.

Plan of Attack

Maygh provides a solution to this problem by making clients cache static resources. Clients load static resources from each other via a central coordinator, which keeps track of the resources cached by each connected client. Since the client code is implemented as a Javascript library loaded by the website, Maygh does not require users to opt in to content distribution.

An outline of the components of the system are provided below. We propose to implement a Javascript client, the coordinator, and a test website that uses our system. If we have time, we will also support multiple coordinators.

- Coordinator
 - Communication with clients
 - Act as protocol server for inter-client communication
 - Handle update requests
 - Handle lookup requests
 - (maybe) Use geo-IP database to determine which client to load from
 - State
 - Content location map
 - Map content to lists of peer-ids
 - Client map
 - Map peer-ids to lists of stored content
 - Upload limits
 - Track bytes downloaded and uploaded
 - Cap upload using upload_ratio
 - Cap upload using upload_max

- (maybe) Multiple coordinators
 - Make content location map distributed using consistent hashing
 - Augment client map to include connected coordinator-id
 - Inter-coordinator communication
 - Initialization
 - Each coordinator should know which content hashes each other coordinator will store
 - lookup requests (whose results need to be cached)
 - Handle lookup requests
 - Use lookup request cache to initialize inter-client communication for clients connected to other coordinators
- Client
 - Persistent storage
 - Communication with other browsers: WebRTC or RTMFP
 - Load content from another client
 - Fallback option if content not found
 - Fallback option if client is malicious or uncooperative
 - Communication with coordinator: WebRTC or RTMFP
 - connection initialization (maygh.connect())
 - update requests
 - lookup requests (maygh.load())
 - requests to connect to another client
- Tests
 - Spoof website with Maygh client code, perhaps using a fake RPC library
 - Single coordinator
 - (maybe) Multiple coordinators
 - (maybe) Malicious clients