Introduction

- Cooperative Caching is the process of a client handling the cache misses of other clients using a central global cache.
- Global Cache knows the clients holding a particular block of data.
- This project introduces Importance-Aware Bloom Filter to cooperative caching algorithm to reduce the global cache requests.

Bloom Filter

- A space-efficient probabilistic data structure that supports set membership queries.
- It uses a boolean array set to F and hash functions to set the bits to T.
- The Bloom Filter supports add and look up.
- The Bloom Filter may return false positives during lookup but no false negatives.
- Importance-Aware Bloom Filter represents important data with high accuracy and deletes less important data.
- The importance is based on the number of times a data occurs.

Framework

Caching Algorithms

Greedy Forwarding

- Each client works in a greedy manner and is not aware of the contents of other clients.
- A client cache miss is handled by other client through the global cache.
- The client receiving the forwarded request provides the data to requesting client.
- If a requested data is not cached by any client, then the request is sent to the server.

N-Chance

- This algorithm does not discard singlets.
- Singlets: Data cached at only one client.
- The singlet information is maintained by the Manager handling global cache.
- The singlets being discarded from a client cache are forwarded to a random client.
- Recirculation Count: Number of times a data can be forwarded.
- This algorithm provides N opportunities before discarding a block from the system.

Robinhood

- This algorithm does not discard singlets.
- The singlets are forwarded to victim clients.
- Victim Client: Client caching Victim Block.
- Victim Block: Block cached at many clients.
- The singlet is forwarded to a victim client and the victim block is replaced with the singlet.
- This algorithm provides a rich block space to a poor block.

Results

Drawbacks

- The number of updates and requests to the global cache reduces the availability of the global cache.
- The client requesting for a block has to wait for the global cache, if an update is being made by other client.
- The global cache receives frequent updates because the global cache has to represent the accurate location of a data block at any point of time.

Proposed Algorithm

- One of the solution to reduce the load of the global cache is to use a less accurate representation of the global cache.
- The less accurate representation uses an array of Importance-Aware Bloom Filter (IBF) each representing the client cache.
- This IBF global cache would serve the requests from client and if it fails then the request is forwarded to the global cache.
- The IBF is a probabilistic view of the client cache and is updated at the IBF global cache at regular intervals.
- The delay in the update of IBF global cache does not affect its performance as it is a probabilistic representation of the client cache and not a perfect representation.

References