A Distributed Web Caching Implementation Using Double Trees

Karan Bhagat kk5408@rit.edu
Advisor: Dr. Minseok Kwon mxkvcs@rit.edu

**Motivation**
Web caching plays an important role in decreasing the perceived latency by the web user while requesting a web object. Even though the web caches bring content closer to the client, this introduces many other problems such as handling load on the web caches in case of a web object popularity, and the replication and replacement strategies. We aim to provide an implementation of web caching which resolves the aforementioned problems.

**Double Trees**
Double Random tree is a k-ary tree data structure for distributed file replica search and propagation. Double trees provide decrease in latency for fetching web objects over its predecessor Random Tree algorithm. It maintains two overlapping trees, one for lookup and other for replication since the lookup queries can be far greater than the replication requests. Each node in the Double Tree is a virtual node mapped to a physical web cache server.

**Distance Aware Double Trees**
- We implement a distributed web caching algorithm by using an altered double tree data structure which can be aware of the web cache distance from the client.
- All the web cache servers are mapped to the virtual nodes in the way such that all the nodes near the root will be farthest to the client and the nodes near leaves will be closest.
- This approach reduces the time for client’s request to reach a random web server since the request will always first reach to one of the double tree’s leaf node, which is closest to the client.

**Tree Construction**
- One double tree is constructed for each web object. Double tree is constructed when a web object is requested for the first time by a client.
- Web caches in a region maintain max heap of all the caches visible to it. This max heap stores the maximum latency to ping a web cache.
- Max heap of web caches can change according to which caches are visible at an instance. The max heap is then used to construct the distance aware double tree.
- Root of the heap is popped and assigned to the double tree root. Then next N nodes in the heap are popped, which are assigned to k children of the root in the double tree. N <= k.

---

**Experiment Setup**

**Results**

![Experiment Setup Diagram]

**REFERENCES**