

## Who Will Share My Load?

The Proxy Server has been designed using tools and frameworks provided in Python. Sockets, BaseHTTPServer, File Handling, urllib2 and MD5 based Hashing Libraries were used in implementing the proposed methodology.

The design involves a Load Balancer which acts as a child server to the Proxy Cluster. The Child Server is exposed to the Clients over the network and acts as gateway to the Parent Servers. The Child Server supports round robin and weighted round robin load balancing techniques. All the requests which is handled by this server is logged to a log file for monitoring the traffic through the Proxy. The load balancer uses multi-threading to serve multiple requests at a given point of time. Based on the load balancing technique, the child server forwards the request to the suitable parent server and fetches back the data associated with the request. Sockets library has been used to facilitate the routing mechanism.

Parent servers accepts the request from the child server and uses a shared caching mechanism. The data associated with the request is either present in the cache which leads to a 'cache hit' or the server has to fetch the data from the source as a consequence of a 'cache miss'. The cache is validated after the request is served in the case of cache hit while the page is fetched from the remote server and is cached. BaseHTTPServer is used to handle requests from the child server along with urllib2 which is used to fetch the pages from the web. Md5 based Hashing Library is used to implement the file storage management.

Team:

Vinayaka Kamath ( [vinayakkamath2010@gmail.com](mailto:vinayakkamath2010@gmail.com) )

01FB16ECS445

Vedartham Bharath ( [vedabharath12345@gmail.com](mailto:vedabharath12345@gmail.com) )

01FB16ECS439

Computer Science and Engineering Department-PES University

## Schematic Diagram:

