Introduction to De-Centralized Architecture

- **What is a De-Centralized Architecture?**

  A **De-Centralized Architecture consist** of N Distributed Servers(DHT) having X number of Peers.The Files from the Peers are sent to the respective servers depending on the hashValue obtained. So in a decentralized system we get a better fault-tolerance as even if one server fails the meta-data replication can easily be got from another DHT server.

- **System Architecture**
  - **Peer(Server+Client)**
    - **HashFunction**
    - **File Download**
  - **Server(DHT)**
    - **Lookup(FileName)**
    - **Register(FileName)**

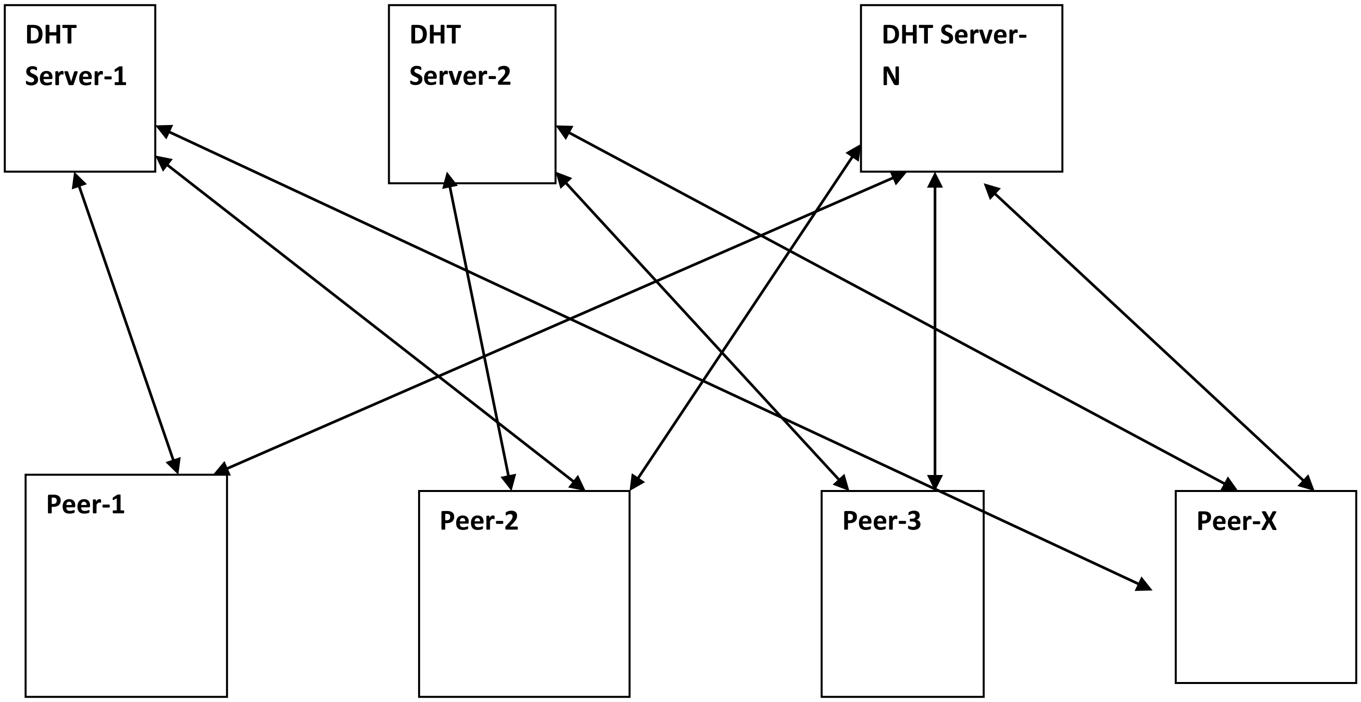
Server(DHT) Architecture

1. The DHT Server stores the value of the file in a hashtable (key:filename,value:IP). The DHT on receiving the FileList maps the fileList on the respective HashMap table. The Server DHT also has a feature to remove the deleted files from its hashMap table.If a file is deleted from the client folder then the on registration the same file is also deleted from the hashMap.

2. The DHT Server returns success if the filename match is found in the hashtable. It also returns the Peers IP and Port address depending on the Replication Factor and on request by the peer.

3. The DHT Server returns the Peer address and Peer Port depending on the replication factor.eg: If the replication factor is 2 it will return the Peer address and Port of 2 other clients who have been connected to the server.The DHT Server will make a random search for getting the Peer Replicas address and Port.

4. File Processing: The DHT Server on receiving the file list will parse the list on the delimiter and store it with the key/value pair formation.



Connection between Server and Peer

Server-Peer Connection for registration and file lookup is denoted by

↔

Peer (Server and Client)

1. The Peer sends the DHT Server the file list it has for registration. When the Peer sends the request to a DHT Server the server stores the file in the HashMap and also sends the backup of the same to a different DHT Server.

2. The Peer when once connected to the DHT Server the connection stays permanent and is stored in a hashtable so when next time a connection is to be established the socket and the outputstreams are retrived from the HashMap file and there is no need to re-establish a connection.

3. File Download takes place between peers the file download size can even support .txt,.bin,.mp4,.mp3 files of sizes of 1KB,1MB upto 1GB.

4. The hash function applied here converts the filename to bytes adds them and on dividing the result by total DHT Serves gives the server name where the file entry has to be stored.

5. Similarly when a file has to be downloaded first for getting the file location a request is send to the DHT Server the Server returns all Peers having the specific file and then a user-selection is required for downloading the file.

6.**Optimized Code:** In the Performance report , reports have been drawn on an optimized code and non-optimized code(base-line code).For optimized code the 10K files are stored in an arraylist depending on the hashvalue so everytime a single message of every filename is not sent to the server but a single message having a group of filenames are sent to the server.So for M(distinct) files having N servers approximately N messages containing (M/N filenames) is send to the DHT.Done to improve performance

7. Non-Optimized Code: This is the baseline of the assignment. Here a single file name is sent to the DHT Server so for M files having Nserver M messages of filename for registeration are sent to the DHT.Provides good response for small number of Filenames.But performance goes down when filenames increased to 10K or more.

Replications

Meta-Data Replication:

- For Meta-Data Replications when the Peer sends its content of FileList to one server it will be simultaneously be send to the next Server as a part of replication.
- So Even if a particular server goes down the data will still be available and the system will be up for functioning.

File Replication

- For replicating Files the files are stored and distributed among the other Peers.
- For File Replication a replication factor is present which decides the number of peers where the file will be replicated.
- A function “Process Replica” is defined which will determine where the files of the folder need to replicated, to which Peer.
- Once the files are replicated and registered from the new peer they can also be downloaded on a request from another Peer.

Future Scope and Enhancements

1. The system must have a more appropriate hashing mechanism and use hash algorithms more accurately . The DHT Server must be able to process Peer request of higher order (>64) concurrently and in a timely efficient manner.

2. Their could be a mechanism to add a failed Server or Peer dynamically back in the network.

3.The File Replication could be done in a more efficient way, by implementing a more robust algorithm so that only the Peer which fails the most frequently its files are replicated.

4. For File Downloads everytime no need to establish a connection ,so the connection can be stored in a List of hashtable.

Conclusion

From the performance evaluation and the design doc it can be concluded that the register operation takes more time then search .The Performance of a Centralized System and a decentralized System almost are the same. The time difference is very low. But there is no time difference between the file Download functionality as it is between the Peers.