**Data Project:**

**IPFS (InterPlanetery File System).**

**Group Members:**

**Faiz Ul Hassan: 22i-0818**

**Hamza Bin Riaz: 22i-1165**

**Muhammad Abdullah: 22i-0820**

About The Project:

IPFS are managed based on content.

In this Project, we have a Ring DHT; a circular Linked List that has all the machines connected. The Program will ask the user the user to enter the identifier Space, then we can select the Order of B Tree. After that,the User Specifies the Identifier Space .Next , the User will get the option to choose personal or random ids for machines which would be inserted in Ring DHT , in increasing order . For Personal Ids , You will select the The ID of the Machine and then you will specify name . And for Random IDs , you only have to specify Machine IDs..

Now a Menu will appear, In which following options will display :

(1).File Insert (4).Machine Insert

(2).File Search (5).Machine Delete

(3).File Delete (6).FT Table Display

**Ring DHT Properties:**

* **Identifier Space:**

Uses a 160-bit circular identifier space based on SHA-1.

Configurable size by the user.

* **Machine and Data Identification:**

Randomly chosen identifiers assigned to machines (nodes) and data (key, value) pairs.

Data with identifier 'e' is stored on the machine 'p' with the smallest identifier 'p ≥ e,' called successor of 'e.'

* **B-Tree Storage:**

Each machine internally stores data using a B-tree.

B-tree indexes keys, and values (files) are stored on the file system.

**Search Algorithm:**

* **Simple Search:**

Originates on any machine.

Locally checks if responsible for the file, else forwards to the next node in the circular linked list.

* **Routing Table Optimization:**

Each machine maintains a routing table with at most O(log N) entries.

Routing tables provide shortcuts/pointers to distant nodes on the circular linked list.

Enhances search efficiency to O(log N).

* **Search Example**:

Illustrated with a search query for key = 12.

Originates at node 28, forwarded through nodes 4, 9, 11, and finally to node 14 for B-tree search.

* **Routing Table Implementation:**

Implemented as doubly linked lists.

Entries maintain the address/pointer of the linked list node corresponding to the machine.

**System Dynamics:**

* **New Machine Joining:**

Calculates H(Machine Name) = p and places the machine in the ring.

Initializes and updates routing tables.

Subdivides Id range between p and succ(p+1), adjusting B-trees accordingly.

**Machine Leaving**:

Assumes graceful departure, redistributing stored files to other machines.

May result in the redistribution of Id ranges and adjustment to B-trees.

**Data Storage Query:**

Can arrive on any machine.

Finds the responsible machine using an algorithm similar to the search query.

**Overall Implementation Notes:**

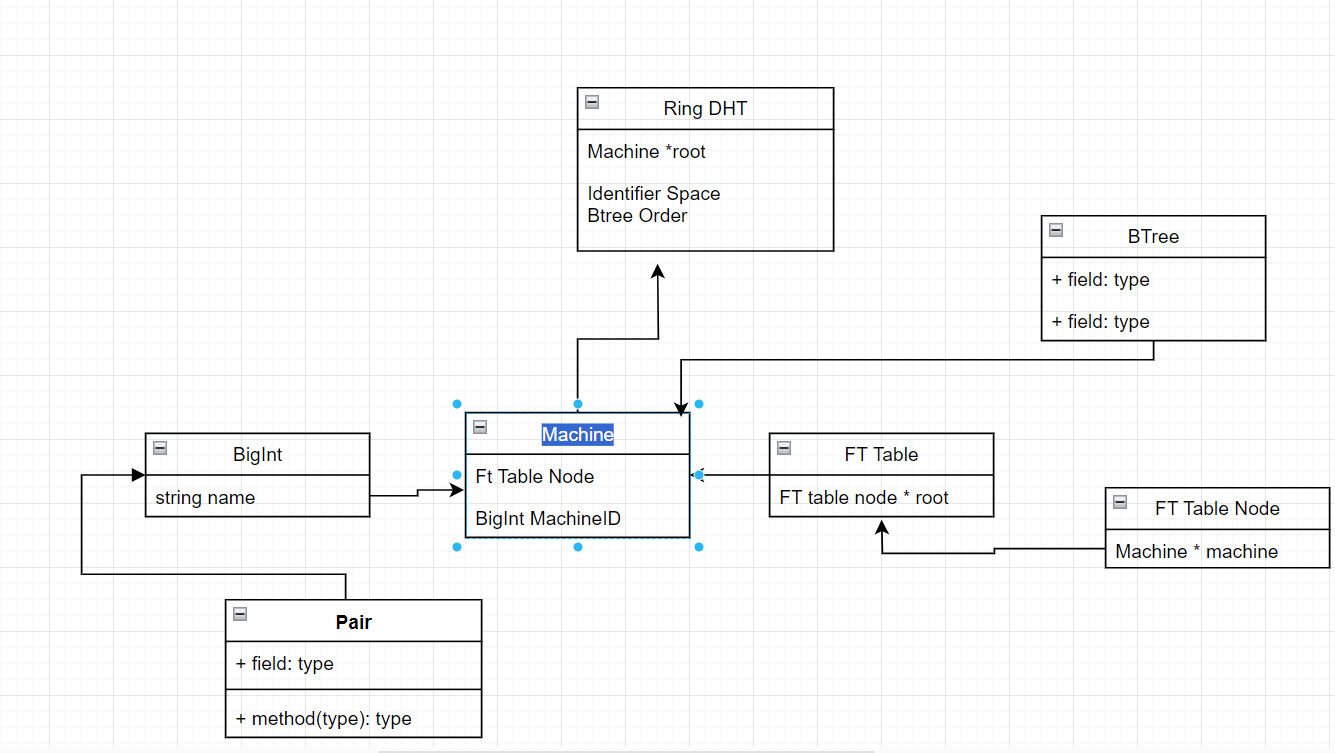
Routing tables are implemented as doubly linked lists.

B-trees index keys, and values (files) are stored on the file system.

Examples provided use simplified scenarios for better understanding.

Circular singly linked list of machines used in the implementation.

**UML DIAGRAM:**

****