

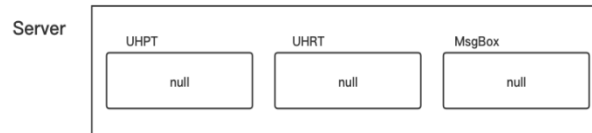
Design

Jinfeng Xu

19206223

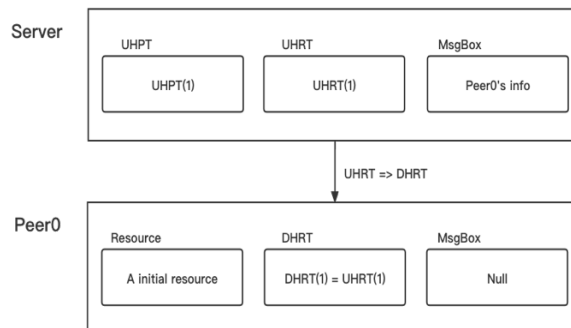
1. Project operation process:

1) Start Server:



The server waits for the peer to connect

2) The first peer is successfully connected:



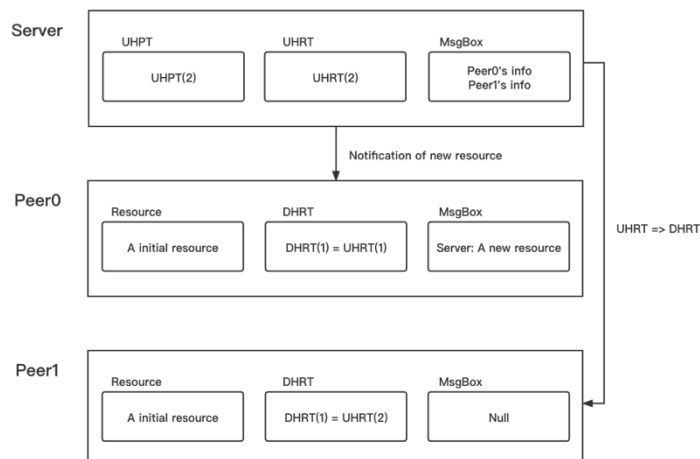
The Server

- <1>. Generate UHRT and UHPT and store them as .txt files.
- <2>. MsgBox Displays basic information about connected peers.
- <3>. Sends the UHRT object at this point to the peer.

The Peer0

- <1>. Receive the UHRT as DHRT and store it as .txt file.
- <2>. Initialize the own resource file.

3) A new peer is connected:



The Server

- <1>. Update UHRT and UHPT and save them as .txt files.
- <2>. MsgBox Displays basic information about connected peers.
- <3>. Sends the current UHRT object to the new peer.
- <4>. Sends notifications of new resources to old peers.

The old Peer

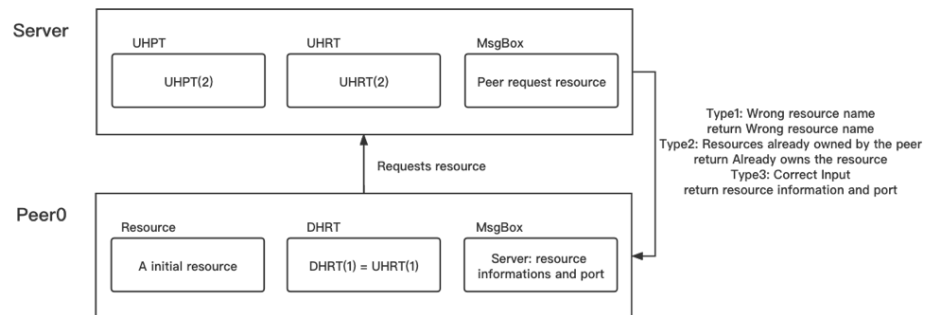
<1>. Received a notification from the server about a new resource

The new Peer

<1>. Receive the UHRT as DHRT and store it as .txt file.

<2>. Initialize the own resource file.

4) The peer requests resource to server:



The Server

<1>. MsgBox Displays Peer request information.

<2>. Verify the Peer request.

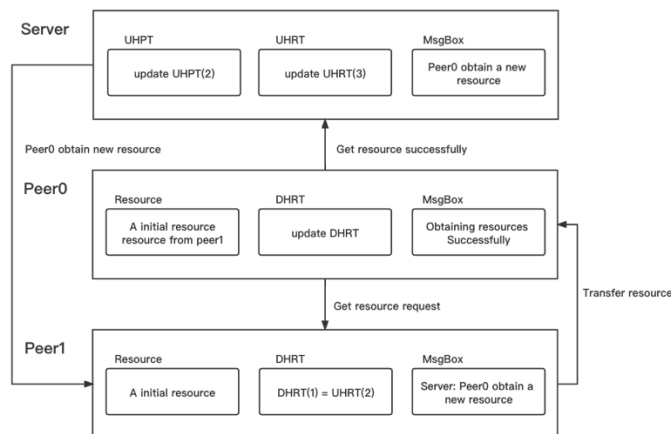
<3>. Return different message notifications based on verification results.

The Peer

<1>. Send a resource retrieval request to the server.

<2>. Receives message feedback from the server.

5) The peer get resource from another peer:



The Server

<1>. Receive the message that Peer0 has successfully obtained the resource and display it in the MsgBox.

<2>. Send resource change notifications to other peers.

<3>. Update UHRT and UHPT.

The Peer0

<1>. Send a request to the peer holding the target resource.

<2>. Accept the resource sent by the peer and save it to itself.

<3>. Update own DHRT.

<4>. Notify the server that the resource has been successfully obtained.

The Peer1

<1>. Receive peer resource request information.

<2>. Send the specified resource to the peer.

<3>. Receive server notifications and display them in MsgBox.

Tips: Corresponding message prompts are provided for all wrong operations.

Include:

Wrong resource name

The requesting peer already owns the resources

Wrong connection address

The connection address points to itself

2. Performance statistics:

1) Peer connects to Server

| Peer | Test1 | Test2 | Test3 | Test4 | Test5 | Test6 | Test7 | Test8 | Test9 | Test10 | Avg |
|--------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Peer1 | 216ms | 213ms | 219ms | 211ms | 216ms | 212ms | 212ms | 221ms | 214ms | 217ms | 215.1ms |
| Peer2 | 214ms | 211ms | 223ms | 214ms | 212ms | 218ms | 211ms | 212ms | 216ms | 222ms | 215.3ms |
| Peer3 | 216ms | 220ms | 209ms | 220ms | 212ms | 221ms | 213ms | 216ms | 211ms | 211ms | 214.9ms |
| Peer4 | 211ms | 217ms | 211ms | 210ms | 217ms | 219ms | 220ms | 216ms | 214ms | 218ms | 215.3ms |
| Peer5 | 219ms | 214ms | 220ms | 215ms | 218ms | 211ms | 212ms | 214ms | 221ms | 210ms | 215.4ms |
| Peer6 | 213ms | 213ms | 215ms | 214ms | 211ms | 217ms | 211ms | 212ms | 213ms | 213ms | 213.2ms |
| Peer7 | 217ms | 217ms | 217ms | 216ms | 218ms | 216ms | 217ms | 217ms | 219ms | 215ms | 216.9ms |
| Peer8 | 217ms | 216ms | 212ms | 215ms | 219ms | 213ms | 214ms | 216ms | 216ms | 215ms | 215.3ms |
| Peer9 | 215ms | 216ms | 217ms | 220ms | 214ms | 214ms | 216ms | 212ms | 216ms | 217ms | 215.7ms |
| Peer10 | 211ms | 216ms | 215ms | 212ms | 217ms | 217ms | 213ms | 219ms | 215ms | 211ms | 214.6ms |
| Avg | 214.9ms | 215.3ms | 215.6ms | 214.7ms | 215.4ms | 215.8ms | 213.9ms | 215.5ms | 215.5ms | 214.8ms | 215.2ms |

The time it takes for the peer to connect to the server is stable at around 215ms

2) Peer connects to Peer

| Peer | Test1 | Test2 | Test3 | Test4 | Test5 | Test6 | Test7 | Test8 | Test9 | Test10 | Avg |
|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|---------|
| Peer | 217ms | 215ms | 213ms | 211ms | 216ms | 214ms | 212ms | 218ms | 214ms | 216ms | 214.6ms |

The time it takes for the peer to connect to the peer is stable at around 215ms

3) Transfer resource

| Size | Test1 | Test2 | Test3 | Test4 | Test5 | Test6 | Test7 | Test8 | Test9 | Test10 | Avg |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----------|
| 10Byte | 1014ms | 1002ms | 1017ms | 1004ms | 1003ms | 1012ms | 1003ms | 1007ms | 1019ms | 1012ms | 1009.3ms |
| 100Byte | 1013ms | 1011ms | 1019ms | 1021ms | 1008ms | 1013ms | 1005ms | 1021ms | 1006ms | 1022ms | 1013.9ms |
| 1KB | 1004ms | 1021ms | 1002ms | 1013ms | 1015ms | 1002ms | 1009ms | 1009ms | 1018ms | 1013ms | 1010.6ms |
| 10KB | 1012ms | 1014ms | 1009ms | 1023ms | 1002ms | 1014ms | 1010ms | 1011ms | 1005ms | 1008ms | 1010.8ms |
| 100KB | 1008ms | 1009ms | 1013ms | 1013ms | 1021ms | 1022ms | 1008ms | 1015ms | 1014ms | 1004ms | 1012.7ms |
| 1MB | 1019ms | 1031ms | 1033ms | 1017ms | 1021ms | 1027ms | 1023ms | 1010ms | 1029ms | 1020ms | 1023.0ms |
| 10MB | 1108ms | 1098ms | 1121ms | 1105ms | 1077ms | 1115ms | 1121ms | 1117ms | 1088ms | 1101ms | 1105.6ms |

For files smaller than 1MB, the transfer time is stable at about 1010ms

For every 1MB increase, the transmission time will increase by about 10ms

3. Pros and Cons:

Advantage:

Achieve all target functions.

Server maintains stable performance even when connected to multiple peers.

The connection between peers is fast and stable

File transfer is fast and stable

With a beautiful and convenient interface

The server has a complete resource notification function to the peer

There are perfect prompts for wrong operations

Highly user-friendly

Disadvantage:

Only support files in .txt format as resources

4. Rationale of design

About UHRT, UHPT, DHRT

These three kinds of DHT are stored by a simple ChordDHT that I implemented. In order to facilitate the query, I use GUID as the key to get the node. GUID has a very low collision probability, and it is encrypted and generated by the SHA-1 algorithm. For UHRT and DHRT, the GUID of the node encrypts the resource name. For UHPT, the GUID of the node is encrypted by the port number of the connected peer. Each node maintains a fingertable. At the same time, the node stores the basic attributes required by the project. Because HRT and HPT share the same node class. The node class generates HRT and HPT resources through constructors with different parameters. In addition to the GUID used to maintain uniqueness, each node also maintains an identifier. Through this identifier and fingertable, a search that is far more efficient than linear can be achieved.

About Server

The main thread of the server maintains the connection of the receiving peer. For each connected peer, a separate thread is opened for communication. UHRT and UHPT are updated when new resources or resource holders change.

About Peer

When the peer connects to the main server, it receives the initialization DHRT and generates its own resources. The main thread of the peer maintains communication with the server. At the same time, after initializing DHRT and its own resources, a new thread is opened to continuously accept resource acquisition requests from other peers. When a resource acquisition request is received, a separate thread is opened for resource transmission, and the thread is closed after completion. DHRT only updates after it acquires new resources.

5. Project structure

```
|—— ChordDHT
|—— Node
|—— Peer
|—— Server
|—— utils
|   |—— SHA1Util //Tools for implementing SHA-1 encryption
|   |—— FileUtil //Tools for file-related operations.
```