

COMP9331 Assignment Report

Fengting YANG

z5089358

Environment

Development environment: Python 3.6.4 on macOS High Sierra (10.13.4)

Test environment: Python 3.6.4 on CSE vLab

How to run the code

Run script:

```
xterm -hold -title "Peer 1" -e "python3 cdht.py 1 3 4" &  
xterm -hold -title "Peer 3" -e "python3 cdht.py 3 4 5" &  
xterm -hold -title "Peer 4" -e "python3 cdht.py 4 5 8" &  
xterm -hold -title "Peer 5" -e "python3 cdht.py 5 8 10" &  
xterm -hold -title "Peer 8" -e "python3 cdht.py 8 10 12" &  
xterm -hold -title "Peer 10" -e "python3 cdht.py 10 12 15" &  
xterm -hold -title "Peer 12" -e "python3 cdht.py 12 15 1" &  
xterm -hold -title "Peer 15" -e "python3 cdht.py 15 1 3" &
```

Input a command:

This CDHT program does not accept any command input since it is hard to implement input and output in one terminal. When it is printing the ping messages, we can't input anything. That's why I write this program to solve this problem. To input a command to peers, use the command.py by typing `python3 command.py` in terminal. Usage: `<peerNumber> <command>[quit|request] <filename>`. Ignore the last parameter if the second parameter is quit. It will send the command and file name to the specified peer.

Design

This program has two threads which are ping thread and TCP thread. Ping thread is to send and receive ping request and response. Also, sequence number and acknowledge number are used in ping thread to check whether a peer is down. In this program, I set the ping interval is 2 seconds, 5 missed packets will cause a peer is recognized as killed. TCP thread is to send and receive complex messages just like file request and quit messages.

Message format

Ping Message (UDP):

{sourcePeerID},{sequenceNumber},{status}

sourcePeerID is the peer which send the ping message.

sequenceNumber is an increase array.

Status has two options: request and response to indicate the message type is a new ping request or the response to a request.

Prompt message and file request message (TCP):

Prompt message:

prompt, {status}, {sourcePeerID}, {succ1}, {succ2}

prompt is to tell the receiver this is a prompt message. It should unpack it as a prompt message format.

Status has three options, querySucc, querySuccResponse and quit. querySucc is to query the successors. querySuccResponse is the sent by the receiver to tell the sender which peer is the successor should be sender's successor. Quit is when a peer leaves gracefully, this peer will send the successor information message to its predecessors in order to keep the circular.

sourcePeerID is the peer which send the ping message.

Succ1 and succ2 are the sender's two successors.

File message:

file, {status}, {sourcePeerID}, {fileName}, {originalPeer}

file is to tell the receiver this is a file message. It should unpack it as a file message format.

Status has three options, here, next and unknown. Here means the sender tell the receiver the file is in sender's peer. Next means the sender tell the receiver the file is in sender's first successor. Unknown means the sender don't know where the file is.

sourcePeerID is the peer which send the ping message.

filename is the request file name.

originalPeer is the peer which request a file.

Code copied from web

killThread function is copied from stackoverflow answered by Johan Dahlin. Site: https://stackoverflow.com/questions/323972/is-there-any-way-to-kill-a-thread-in-python?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

Demo

Demo site: <https://youtu.be/EOMFy3L0Lrk>