

MOVIE SHARE APP

MOVIE RATING, QUERYING AND CHATting APP

By: Lillian Huang, Haoxue Li, Chenhao Pan,
Xuefei Zhou

TABLE OF CONTENTS

01

ABOUT OUR APP

What is our
application about?

02

PEERBASE

P2P Framework
explained

03

CORE PACKAGE

Core Package
explained

04

HANDLER PACKAGE

Handler Package
explained

05

GUI PACKAGE

GUI Package
explained

06

LISTENER PACKAGE

Listener Package
explained

07

PROBLEMS

Problems
Encountered in
Coding Process

08

LIVE DEMO

Demo of Movie
Sharing Application

01 ABOUT OUR APP

What is our application
about?



ABOUT OUR APP

- Motivation for the application:
 - Rotten Tomatoes
 - IMDb Movies
- The Movie Share Application is built on the PeerBase P2P framework. A node may have a movie database read capability or read/write capability. There is a configurable maximum number of peers for each node.
- The application has three main functionalities:
 - Query a movie
 - Rate a movie
 - Send a message.

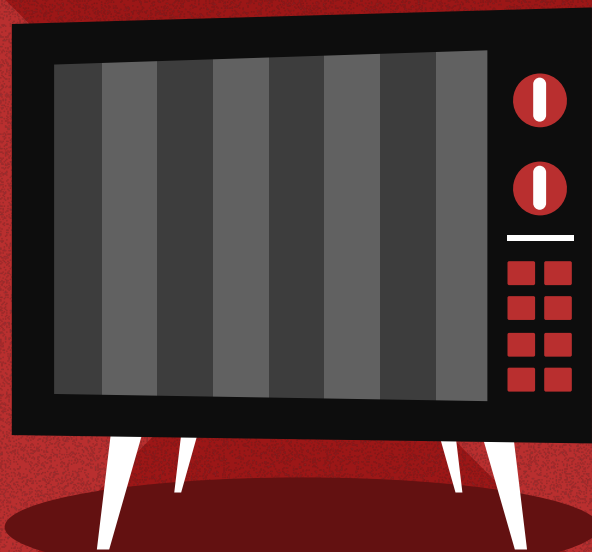


02 PEERBASE FRAMEWORK

P2P Framework Explained



P2P FRAMEWORK



- The PeerBase P2P framework was built by Nadeem Abdul Hamid.
- The author used the network sockets to build two-way communication between two network nodes.
- A node discovers and builds peer nodes by probing the peer nodes of its known peers.
- The probing is limited by a configurable parameter of maximum hops.
- The PeerBase has a stabilizer mechanism.

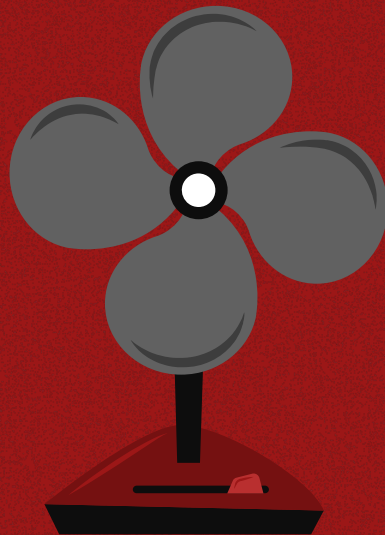
PACKAGES AND FILES IN THE P2P FRAMEWORK

- Socket Package
 - NormalSocket
 - NormalSocketFactory
 - SocketFactory
 - SocketInterface
- Util Package
 - SimplePingStabilizer
 - SimpleRouter
- HandlerInterface
- LoggerUtil
- Node
 - PeerHandler (Inner Class)
 - StabilizerRunner (Inner Class)
- PeerConnection
- PeerInfo
- PeerMessage
- RouterInterface
- StabilizerInterface



03 CORE PACKAGE

Core Package Explained



FILES IN THE CORE PACKAGE

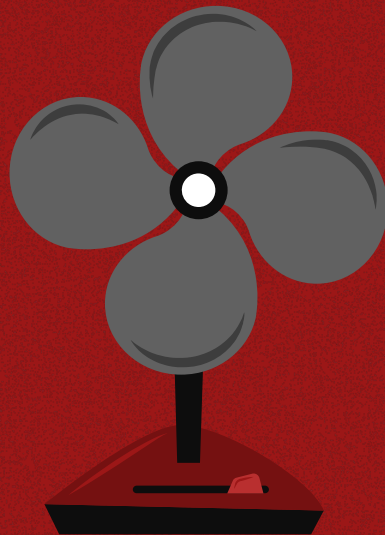
- Constants
 - Define message types
- MovieShareNode
 - Router (Inner Class)
 - This node class extends the Node class from the PeerBase Framework.
 - Thorough explanation on this class will be provided in the next slide
- MovieShareService
 - This class initialize the Movie Share application and starts the peer node
- MovieShareStabilizer
 - This class extends the PeerBase stabilizer. It detects dead peer nodes and removes them from the peer list. It also updates the peer list display on the UI.
- NodeDBAccess
 - This class contains database connection information.

MOVIE SHARE NODE EXPLAINED

- This class contains the following main functions:
 - Build Peers
 - Update Peer List Display
 - Router (Inner class)
- Build Peers:
 - This function builds the peer list of a node by contacting its neighbor peer and expanding recursively within its configured maximum peer hops and also configured maximum peers.
- Update Peer List Display:
 - This function updates the peer list on the UI when the peer list is updated (either add a new peer or remove a peer)
- Router
 - This class finds a peer to route a message.

04 HANDLER PACKAGE

Handler Package Explained



FILES IN THE HANDLER PACKAGE

- NameHandler
- JoinHandler
- ListHandler
- QueryHandler
- QueryProcessor
- QueryResponseHandler
- RateHandler
- RateProcessor
- RateResponseHandler
- MessageHandler



EXPLANATIONS OF HANDLERS

- Name Handler:
 - This handler is used to get the node id from the neighbor node
 - Used during build peers
- Join Handler
 - This handler is used when inserting nodes into a peerlist
 - Used during build peers
- List Handler
 - This handler loops through its peer node list and sends all peer list information to the requesting node.
 - Used also during build peers
- Query Handler
 - This handler is used when querying for movies
 - Sends information to the Query Processor which will process the query
- Query Processor
 - Accesses the database to get query result and returns the result using a QueryResponseHandler
- QueryResponseHandler
 - Sends the result back to the node requesting the query and updates GUI output
 - If the responding node is not already in the peer list, it will be added.

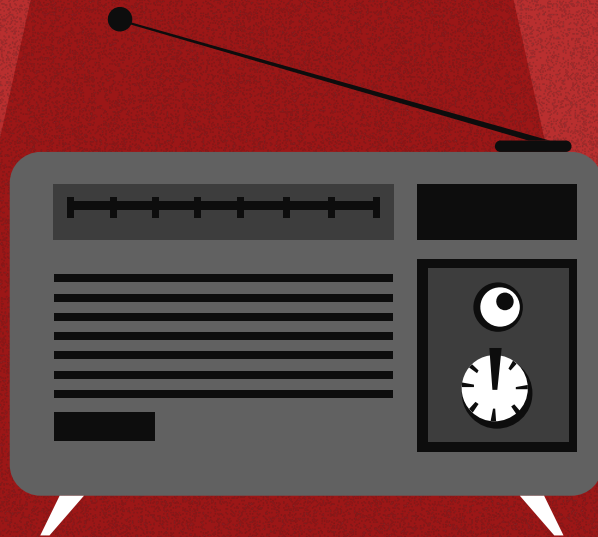
EXPLANATION OF HANDLERS (CONT.)

- RateHandler
 - This handler is used when rating for movies
 - Sends information to the Rate Processor which will process the rating request
- RateProcessor
 - Accesses the database to rate the movie and returns the updated rating using a RateResponseHandler
- RateResponseHandler
 - Sends the result back to the node requesting the query and updates GUI output
 - If peer who responded to the request is not in existing peer list, it will be added.
- MessageHandler
 - This handler is used when messaging peers.
 - Sends the message to all peers in the peerlist and other nodes within its configured maximum message hops

05

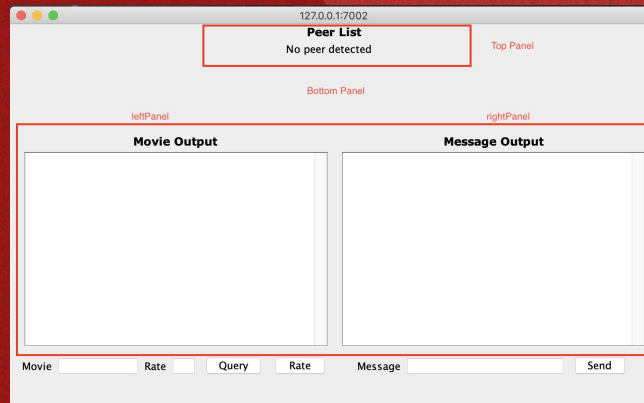
GUI PACKAGE

GUI Package Explained



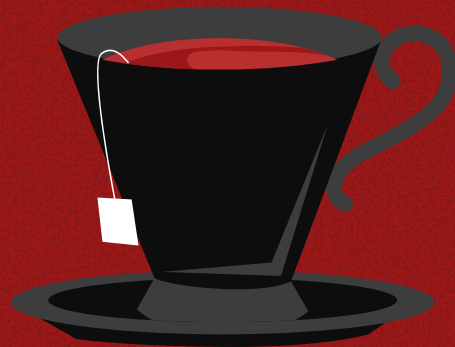
FILES IN GUI PACKAGE AND EXPLANATIONS

- GUI Display
 - JFrame with title with node id (host and port)
 - Top panel shows the peerlist of the node
 - Bottom panel includes left panel and right panel
 - Left panel controls all movie related requests
 - Query and Rate Buttons
 - Right panel controls all message related requests
 - Send Button



06 LISTENER PACKAGE

Listener Package Explained



FILES IN LISTENER PACKAGE

- **MovieQueryButtonListener**
 - Get text from movie input box in the GUI display
 - Creates message data
 - Sends the message to the node's peers to handle the query request
 - Clears the input box in the GUI display
- **RateMovieButtonListener**
 - Get text from movie input box in the GUI display
 - Get text from the rating input box in the GUI display
 - Creates message data
 - Sends the message to the node's peers to handle the rate request
 - Clears the input boxes in the GUI display
- **MessageButtonListener**
 - Get text from the message input box in the GUI display
 - Creates the message data
 - Sends the message to the node's peers
 - Clears the input box in the GUI display

07

PROBLEMS

Problems Encountered in
Coding



PROBLEMS ENCOUNTERED IN CODING PROCESS

- **Problem:** Multiple responses can be received per request
 - multiple peer nodes send results back to the sender.
- **Solution:**
 - Decided to add a duplicate detection mechanism
 - Introduced a message id to indicate the message sequence number for each message type
 - At the receiving node, it records the message id it has processed.
 - A message with message id less or equal to the recorded id will be ignored.
- In some cases, we still receive duplicate messages
 - PeerBase framework uses multithreading when it handles incoming messages which can cause race conditions when the node receives the same messages from different peers.
 - Such duplicate message problems are difficult to resolve unless we change the PeerBase framework.



08

LIVE DEMO

HOW THE APPLICATION RUNS

- Once the application is running, it will first read the properties file for the information regarding:
 - Database information (database name, username, password, server, driver), database read-write access, maximum number of peers for this node, how many hops are allowed when building peers, how many hops are allowed when querying for movies, how many hops are allowed when messaging peers, node's host and port, neighbor node's host and port
- It will make a new service and pass in the properties as a parameter
- The service will then be initialized by :
 - Creating PeerInfo for the node and it's neighbor
 - Creating a database connection
 - Initializing the node where message handlers are initialized in the constructor
 - Initialize the GUI display and add button Action Listeners
 - Building peers for the node
 - Starting a thread to run the main loop for the node, in which it open a server socket to listen for incoming messages. When a message comes, it looks up a handler to process the message.
 - Creating a thread to run a stabilizer to detect dead peer nodes.

SCENARIO 1

Node Id(Host:Port)	Neighbor Node Id	Read Access	Read/Write Access
127.0.0.1:7001	127.0.0.1:7002	F	F
127.0.0.1:7002	127.0.0.1:7003	T	F
127.0.0.1:7003	127.0.0.1:7004	T	T
127.0.0.1:7004	127.0.0.1:7003	T	T

SCENARIO 2

Node Id(Host:Port)	Neighbor Node Id	Read Access	Read/Write Access
127.0.0.1:7001	127.0.0.1:7002	F	F
127.0.0.1:7002	127.0.0.1:7001	F	F

SCENARIO 3

Node Id(Host:Port)	Neighbor Node Id	Read Access	Read/Write Access
127.0.0.1:7001	127.0.0.1:7002	F	F
127.0.0.1:7002	127.0.0.1:7003	T	T
127.0.0.1:7003	127.0.0.1:7004	F	F
127.0.0.1:7004	127.0.0.1:7001	T	T

SCENARIO 4

Node Id(Host:Port)	Neighbor Node Id	Read Access	Read/Write Access
192.168.86.88:7003	192.168.86.135:7001	F	F
192.168.86.135:7001	192.168.86.88:7003	T	T



THANKS!

Do you have any questions?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution

