Make a database connection and retrieve records into class variables (arrays)

Make a hasmap that contains a mapping between unique from addresses and a vector of messages sent by them

Make a Thread Pool of size=size of hashmap (Pool 1)

Make worker threads and add them to pool 1

Execute them until completion of all thread in a parallel fashion

Set host, username, and password and enable tls

Make a session variable for authentication

Create a message packet for the session and authenticate the credentials with host (smtp connection) and save the connections

Make a Thread Pool of size=no of records in database (Pool 2)

Make message worker threads and add them to pool 2

Execute them until completion of all thread in a parallel fashion

Use the saved connections made by worker threads in pool 1

Set the message headers (from,to,subject) and body of the message

Send the message

Close the smtp connections made for the different from addresses

Approach

- First threads are run to make as many smtp connections as there are unique from addresses in parallel
- These connections are saved to be used to send all the messages in the records in the database (one connection is made per from address to send all messages belonging to it)
- Next using these limited connections messages are sent by threads in a parallel fashion
- Number of connections are greatly reduced in this approach.

Advantages

Only one connection is made per from address so no message will be rejected by smtp server (that was with the case of making multiple smtp logins). Message delivery is guaranteed

Disadvantage

Little slower message delivery as smtp servers don't allow sending of messages by a single connection in a parallel fashions. (Messages are queued at the sender's smtp server before sending)