O-PAD: Collaborative Text Editor

Dushyant Panchal | Osheen Sachdev | Paras Mehan | Adwit Singh Kochar
2018033 2018059 2018062 2018276

Problem Statement

Distributed text editors are commonly used to support collaborative editing for software or document development. However, many existing solutions work with documents on the server; they do not support direct collaboration between two clients working with locally stored documents. We propose a distributed text editor for real-time collaboration.

Implementation

File Tracker: Maintains a map of files and the list of online O-Pad clients working on that file, i.e. the collaborators. Any new client C1 interested in editing file F contacts the file tracker to get the address of one of the online O-Pad clients, say C2 working on F. C1 can then request C2 for the latest version of F.
O-Pad Client: O-Pad client is the application running on the user side that manages the text editor and sends and receives updates regarding changed files.
Message Broker: Maintains queues for each file being edited live. An update made to F by client C is pushed as a publish message to the queue of F. Rest of the collaborators of F consume the message to update their copy.

If Users 1, 2 and 3 want to edit a file simultaneously, and they open O-Pad in processes O-Pad1, O-Pad2, O-Pad3, respectively (all on different machines)

1. User 1 will create a new file F1
    a. O-Pad1 requests the file tracker using **ZeroMQ** to create an entry for F1 and add itself to the online users of F1.
    b. O-Pad1 creates a new queue Q1 in **RabbitMQ** and subscribes to it
2. User 2 requests to join F1
    a. O-Pad2 subscribes to Q1
    b. O-Pad2 sends a request to the file tracker using **ZeroMQ** to fetch the current version of F1 and add itself to the online users of F1.
    c. The file tracker responds by giving the address of one of the online collaborators (e.g. O-Pad1)
    d. O-Pad2 requests O-Pad1 to send F1.

   e. O-Pad1 replies with the file to O-Pad2, also containing the last modified timestamp.
   f. O-Pad2 applies all the changes received from Q1 after the last modified timestamp.
3. User 1 makes changes to F1.
   a. O-Pad1 notes the changes and publishes a message containing the changes to Q1.
   b. O-Pad2 receives the update message from Q1.
   c. O-Pad2 updates the opened file according to User 1's updates
4. User 1 and User 2 make an update concurrently
   a. A global ordering is decided between User 1's update and User 2's update, and according to the order, O-Pad1 and two will update the local files.

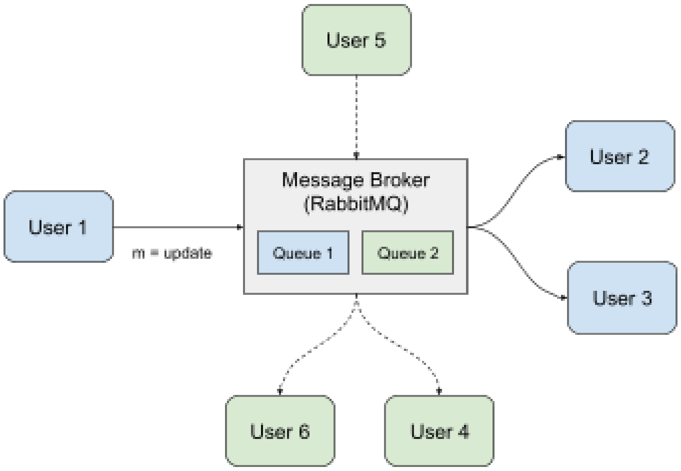

Fig 1: Multiple queues for different files

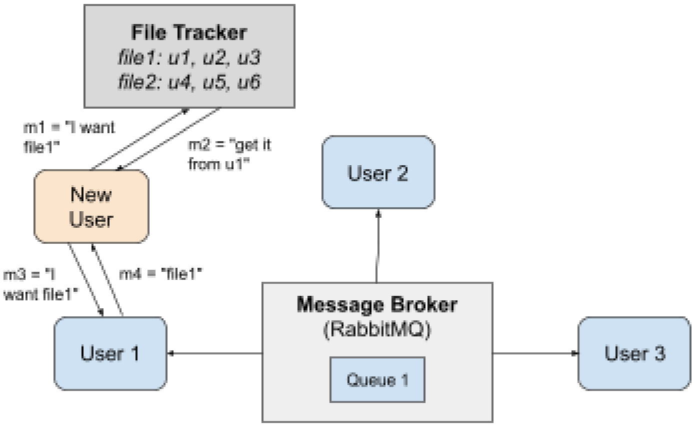

Fig 2: Joining of a new user

Final Outcome

1. An application where users can collaborate with others on text files.
2. Users can create new text files or start collaborating on existing files.
3. The text files won't be stored on any central server but instead saved on the user's side.