CS 632 A: Distributed Editing

November 14, 2014

Group A:

- Arihant Jain
- Bhavishya Mittal

Group B:

- Ashok Kumar
- Javesh Garg

What is this all about?

- Develop a peer to peer scalable collaborative text editor
- Understanding and implementing distributed operational transforms
- Handle all basic operations like insert, delete and show the changes in real time to all collaborators
- Speed and reliability are most crucial points to take into account

Why is this so complex?

- Peer to peer model and not a client-server
- Unlike OT in client-server model, in peer-to-peer model, there is no server to resolve conflicts. All the decisions have to be taken by each client in a distributed fashion
- No single file at one place, each collaborator has own copy which needs to be synced with others in real time
- Handle real time joining or leaving by some nodes

Similarities

- Peer to Peer
- **BSS algorithm** to implement distributed operational transforms since a change needs to be broadcasted to all other clients.
- PyQt4: implemented the front end and GUI using python framework
- Minimum load at document index server and communicate with it only when a new node joins or leaves.
- Use one to all broadcast for changes at all nodes in $O(log(n))$ time.

Differences

- **Overlay Network Used**
  - **Group A**: DHT Chordal Ring and n-ary tree
  - **Group B**: Complete graph
- **Server Role**
  - **Group A**: Server stores just one IP corresponding to each distinct document. Server handles joining or leaving of a node one at a time(constraint due to chordal ring)
  - **Group B**: Server stores IPs of all collaborators for each document. Server can never be a bottleneck as it handles each new communication in a newly spun thread
- **When new node arrives**
  - **Group A**: All pending changes are applied and copy send to the new client via indexing server.
  - **Group B**: the new collaborator asks for a copy of document from one of the current collaborators only and no role of server

Differences

- **Functionalities**
  - **Group A**: Import, save, list of all currently opened docs, print to pdf, alignment, formatting - highlight, font - color, size, family, print preview
  - **Group B**: Plain-text editor
- **Merging documents**
  - **Group A**: conflict resolution based on IDs of the nodes at the time of OT in real time, hence, no surplus work for any client.
  - **Group B**: merge files after a regular interval of time to resolve any conflict. node 0 is made leader and sends changes to other nodes in $O(log(n))$ time.

Improvements

- Inserting features like images, graphs, tables
- Handling abrupt node failure by ping
- Further improvements in GUI
- History of edits made available to all collaborators