CPSC538B Draft Proposal

Idea: P2P Wikipedia

Specifically: Information retrieval and document linking in P2P networks

Overall Goal

Create a P2P app that lets you read, create, revise and delete Wikipedia style documents. The topology should be 100% P2P with an strong emphasis on fast document retrieval (by maintaining some global state) and semantics (how to locate resources -- hyperlink analogue). Secondary concerns include document availability (replication), revision conflicts while tertiary concerns are custom document storage.

Our project will address the questions

1. How do we know what documents exists in the network? How is this info maintained when documents are created/deleted?
2. How do we locate the node with the specific document we want? What if it is down? Can we make a performance guarantee for document retrieval?
3. How to we guarantee that we are viewing the latest revision of the document?
4. How do we guarantee that revisions are replicated? What about creations/deletes?
5. What level of fault-tolerance should we accept (with regards to document availability)?
6. What is the equivalent to hyperlinks in a P2P network?
7. What kind of mechanisms can be put is place to prevent malicious users from changing content? (Maybe, users have the option of seeing previous revisions if they don't like the newest version)
8. How do we communicate among nodes?

Focus

1. Fast document retrieval
2. Linking documents (more generally, linking content in a peer-to-peer network in a stable way)

Challenges

1. Searching for documents
2. Linking documents (some analogue of hyperlinks)
3. Global state management
4. Document conflicts
5. Protocol to support the above points

Assumptions

Implementation

1. Document titles are unique
2. Once node with desired document is found, use client/server model with HTTP
3. The part of server component of app is already done (i.e. CouchDB)

System

1. Nodes are non-Byzantine (may relax: relates to point (7) in questions section)
2. A node's IP address doesn't change (may relax: relates to point (8) in questions section)
3. Packets are correctly ordered and aren't lost

Tasks

Research

1. Distributed hash tables (DHT), Chord and Content Addressable Networks (CAN)
2. Semantic peer-to-peer networks
3. Outline protocol (how to resolve ID to IP address for http; how to initiate a search)

Programming - Pass 1

1. Implement "fast document retrieval" algorithm in GO (possibly a DHT table)
2. Implement app protocol (to get address info -- actual communication via http/REST)
3. App UI (at least to view documents). I expect that this will largely just be using an html library for GO.
4. Setup up CouchDB.

Programming - Pass 2

1. Implement performance monitoring + log visualization
2. Replace CouchDB with own implementation (if required, time permitting)