CS6675/ CS4675

Homework Assignment 2
(Programming Category)

Student Name:______________________

Student Session: cs6675 or CS4675 (circle one)

You are given the choice of two types of programming problems in the first homework assignment. Each programming problem consists of multiple options.

You only need to choose one option from one of the problems as your first homework.

Feel free to choose any of your favorite programming language Java, C, Perl, Python.

Due Date: Midnight on Friday of Week 3 (Jan 24) with graceful (no penalty) extension till 9am on Jan 25 (Sat). **No late submission will be accepted.**

Problem 1. Hand-on Experience with a Web Crawler

(1) You are asked to read the article on Web Crawler at https://en.wikipedia.org/wiki/Web_crawler
(2) You are asked install and run an open source crawler, which can be found in some open source search engine software, such as Apache Lucene search engine (http://lucene.apache.org/core/), and report your experience.

This problem has a number of options. You are expected to perform one of the following two options:

Option 1.1: Experience with an Open Source Crawlers

(1) Download and install a Web Crawler, such as an open source crawler which can be found from https://en.wikipedia.org/wiki/Web_crawler or PeerCrawl (http://www.cc.gatech.edu/projects/disl/PeerCrawl/).
(2) Select a seed URL to initialize your Web crawler, such as cc.gatech.edu, and you are expected to crawl at least 1000 URLs.
(3) Show the design of your Web archive to store your crawled web pages, the keywords (subjects) you have extracted.
(4) Plot the crawl speed in terms of the number of keywords you have extracted and the number of URLs you have extracted and the number of URLs you are able to crawl.
(5) Discuss your experience and lessons learned. Predict how long your crawler may need to work in order to crawl 10 millions of pages and 1 billion of pages.

Deliverable:
(a) Source code and executable with readme.
(b) ScreenShots of your Web Crawler's Command lines or GUIs

(c) Crawl Statistics (Crawl speed→ #pages/minute, ratio of #URL crawled / #URL to be crawled, etc.) in excel plots or tabular format
(d) Discuss your experience and lessons learned.

Option 1.2: Write a Web Crawler of your own.
Feel free to use any open source crawler as the code base. Write your focused crawler, such as crawling only CoC website or crawling only healthcare web pages. You are expected to crawl at least 1000 pages.

Deliverable:
(a) Source code and executable with readme.
(b) Discuss the design of your crawler: Pros and cons.
(c) ScreenShots of your Web Crawler's Command lines or GUIs
(d) Crawl Statistics (Crawl speed→ #pages/minute, ratio of #URL crawled / #URL to be crawled, etc.) in excel plots or tabular format
(e) Discuss your experience and lessons learned.

Problem 2. Hand-on Experience with Open Source Peer to Peer System

This programming assignment will help you get h**and-on experience with an open source Peer to Peer system.**

First, you are asked to download an open source peer to peer software system.

Here are some example P2P systems:

- Gnutella: http://sourceforge.net/projects/gtk-gnutella/
- PeerSim: A P2P Simulator: http://peersim.sourceforge.net
- OpenVoIP: http://www1.cs.columbia.edu/~salman/peer/
- Morpheus: http://cwenger.github.io/Morpheus/
- Bittorrent((http://www.bittorrent.com), read at http://lifehacker.com/285489/a-beginners-guide-to-bittorrent
- Chord: http://www.vromans.org/johan/projects/Chordii/
- MemCached: a peer to peer cache system (http://memcached.org, https://pecl.php.net/package/memcache).
- Open Chord (https://sourceforge.net/projects/open-chord/);
- OpenDHT (https://github.com/savoirfairelinux/opendht/blob/master/README.md)
- Feel free to use your most favorite P2P system instead of anyone from the above list.

You are expected to create a toy peer to peer network system with a minimum of 5 peers. Ideally, each peer has a skewed degree distribution (e.g., may connect to different number of other nodes).

Most of the P2P package downloaded will provide an example workload such as keyword based file/message sharing. You may also create a P2P application of your own, such as sharing files or songs or images, videos. You are allowed to use open source P2P file sharing package to setup a toy P2P file sharing.

When you download a P2P client package, you have two options to set it up:

(1) You can create multiple clients on your laptop and enable them to communicate with one another by joining the P2P network. This creates a simulated toy P2P overlay under your control.

(2) You can also setup your P2P client as instructed and join the full fledged P2P network with the default application. In this case, you are a user (client) and a routing node or a content server of this P2P network.

There are three options under this second problem. The option 1 and option 2 are designed for the beginners.

Option 1 (For beginners of Peer to Peer systems):

Once you complete the setup of your toy P2P search system with minimum of 5 peers. You are asked to install 5 song files or text files at each peer. Hint: search keywords should be used to name your files. Then perform two types of measurement for your toy P2P search system.

(1) Baseline measurement: you are asked to measure the performance of your keyword query requests in terms of the latency (average time needed per query) and throughput (#queries served per time unit).

(2) You are required to use one of the following three types of workload setups:

(2.1) varying the number of files each peer is shared from 5 to 10, 20 files among 5 peers with a fixed number of test queries, say 10. Compare the latency and throughput performance of these three top P2P systems.

(2.2) varying the number of queries from 10, 20, 40, but with a fixed number of files shared among the 5 peers, say 5 or 10. Compare the latency and throughput performance of these three top P2P systems.

(2.3) varying the number of peers in your toy system from 5 to 10, 20. Compare the latency and throughput performance of these three toy P2P systems.

Deliverable:

- URL of the P2P source code downloaded.
- Screen shots of your P2P Command lines or GUIs, showing the membership, the query and the routing functionality (operations) of this toy P2P system.
- Measurement comparison in terms of throughput and latency of two routing protocols of this toy P2P system: the default routing protocol and your proposed routing protocol.

- Discuss latency and throughput of your toy P2P system under the two alternative routing protocols in terms of pros and cons.
- Discuss scalability, reliability and anonymity of your P2P system.
- [Option] Discuss any additional functionality you wish to introduce if any.

Option 2 (For beginners of Peer to Peer systems):

You are expected to setup your P2P client as instructed from your favorite P2P package and join the full fledge P2P network with the pre-defined (default) application. In this case, you are a user (client) and a routing node or a content server of this P2P network. You can send query to the network and measure the performance of your queries in terms of the latency (average time needed per query) and throughput (#queries served per time unit, e.g., per minute).

Your measurements should be done by varying the number of queries from 10, 20, 40 and compare the latency and throughput performance of these three workloads.

Finally, you are asked to measure the cost of hand-shake messages of your client, which are the ping and pong messages to keep your client connected to the P2P network by hand-shaking with your neighbor peers.

Optional task (bonus but not required): You may also add some optimization techniques to improve the search performance, such as caching on your local client by modifying the client code running on your laptop. Then compare your modified client with the original client to show for the same set of queries, your modified client can reduce query latency when caching is being utilized.

Deliverable:

- URL of the P2P source code downloaded.
- Screen shots of your P2P Command lines or GUIs, showing the membership, the query and the routing functionality (operations) of this toy P2P system.
- Run measurements by designing queries to be sent out from your P2P client to the overlay network and measure query latency and throughput of your client.
    - You may have keyword queries that will obtain results and also dummy queries that should not have any meaningful results to be returned.
- In addition to designing and measuring queries, also measuring the membership establishment and membership maintenance cost, measuring P2P dynamics such as the changing frequency of neighboring peers from the eye of your P2P Client. You can measure one-hop dynamics, two hop dynamics, and so forth. Provide measurement results and analysis on three interesting measurement based observations about the P2P overlay from your P2P client perspective.
- Use your imagination to report three interesting observations you made from your measurements.
- [Option] Discuss any additional functionality you may want to introduce if any.

- [Bonus task] In case that you modified your client software, measure both modified client and original client and compare the measurement results. Provide analysis of the comparison results on latency and throughput of your toy P2P system when using the original client to serve queries v.s. using the local cache based optimization or any other types of optimizations, and show the pros and cons of your proposal.

Option 3: This option is designed for students who have some experiences with P2P systems and want to gain hand-on experience with advanced Peer to Peer System.

Once you complete the setup of your toy P2P system with minimum of 5 peers. You are asked to improve the routing protocol of your toy P2P system by proposing your own peer to peer routing protocol. Take unstructured P2P as an example, you may design a friend based routing, a short cut based routing, or a random walk based routing protocol or other semantic based routing. Then compare your routing protocol with the default broadcast routing using P2P search queries over the files shared among the 5 peers. Discuss your comparison results.

For measurement comparison of your proposed optimization with the baseline broadcast protocol, you are required to use one of the following two workload setups:

(1) sharing 10, 20, 40 files among peers with a fixed number of test queries, say 5.

(2) varying the number of queries from 5, 10, to 20 and a fixed number of files shared among the 5 peers, say 10. You may consider each keyword request for files shared as one transaction and measure each transaction's latency and measure per minute throughput (#transactions done).

Deliverable:

- URL of the P2P source code downloaded.
- Screen shots of your P2P Command lines or GUIs, showing the membership, the query and the routing functionality (operations) of this toy P2P system.
- Measurement comparison in terms of throughput and latency of two routing protocols of this toy P2P system: the default routing protocol and your proposed routing protocol.
- Discuss latency and throughput of your toy P2P system under the two alternative routing protocols in terms of pros and cons.
- [Option] Discuss any additional functionality you may want to introduce if any.