Project TiniTorrent: Phase One

Javed I. Khan
Fall 2013, Deadline: October 31, 2013
ST: 6/759995 Foundations of Peer-to-Peer Computing
Department of Computer Science, Kent State University
javed@kent.edu

Before you start the first phase make sure you have read the related documents and could answer the TiniTorrent roadmap questions. The assignment has steps. Each part involves some programming, few documentation, and some experimentation. You need to solve the implementation problem to perform the experimentation part. Each part has specific points. Your project deliverables are (a) a zip file with the project directory (with source code, makefile, sample data file, etc.) and (b) a report (in MS Word format) which should contain answers to both the required documentation and experimentation results.

1.1 Assignment-1: NoniTorrent: (200+0+0=200 points)

Problem: We start the project in client server which would exchange a file in one piece. Take the server.c and convert it to a seeder.c and the client.c into a leecher.c. Let the leecher obtain the sample file from the seeder as one large file. Use TiniTorrent messages to coordinate their actions.

Documentation: None.

Experiment: Run the two clients at two hosts and show that the files do transfer to the leecher to yourself.

1.2 Assignment-2: Client & Server IniTorrent: (400+0+600=800 points)

Problem: Now complete the design of the leecher and seeder so that the file is split into ***p*** (p= 32) pieces- each of ***B*** (B=512 bytes) and the inileecher.c can fetch the file piece by piece from the iniseeder.c. (Now you should be able to run this final program in both modes by setting the piece size parameter in myinit file.). The parameters *filename*, *p* and *B* can be entered as command line argument.

Now make sure you can run both components on GENI. Now prepare for the GENI experiments described below:

Documentation: None.

Experiment: (a) Start the programs on two hosts and run them in whole file mode. Note the completion time. (b) Now start the programs in pieces mode. Seed the seeder with all p=32 pieces, B=512 bytes and the leecher with none. Trace the piece receipt time at the leecher and plot it (y-axis) against piece indices (x-axis). Describe the plot. Note the overall completion time. (c) Repeat the experiment with 2-8 leechers and one seed. Compare the overall completion time (x-axis) with the number of leechers (y-axis). Explain the plot. (You may need to place some code in the peer so that you can all required measure time, and let all of them may start at the same time).

1.3 Assignment-3: P2P TiniTorrent: (100+300+600= 1000 points)

Problem: Join seeder.c and leecher.c into one program with two modules and build tinitorrent.c. This should be able to act as both server and client. You may want to use threads to separate them.

Documentation: (a) Explain briefly the design. (b) Draw and explain the program flowcharts. (c) Draw and explain the state table data structure or CARDs. (d) Draw and explain the message sequences between the seeder and the leechers.

Experiment: (a) Start two instances of TiniTorrent on two hosts. Evenly seed them (seed odd pieces in one and even pieces in the other). Let them now exchange the pieces. Trace the piece receipt time and plot it (y-axis) against piece

indices (x-axis) (Fig-3.1). Describe the plot. Note the swarm completion time (b) Repeat the experiment with more (3-8) peer instances. Compare the swarm completion time (x-axis) the number of peers (y-axis) in the swarm. Explain the plot (Fig-3.2). (c) Now use a swarm of size 4 and repeat the experiment where the number of pieces p is varied from 128, 64, 32 and 16. Plot the swarm completion time with respect to number of pieces (x-axis) (Fig-3.3). Explain the graph.

1.4 Assignment-4: P2P TiniTorrent: (200+200+600=1000 points)

Problem: Modify the TiniTorrent so that each exchange slave can handle multiple pieces over the same connection. A receiving peer should be able to send more requests over the same connection when sending peer has other pieces. A sending peer should be able to place counter requests for other pieces. The receiving peer who is currently receiving should be able to honor such counter requests. Also include print statement in your peer so it can create a log/trace of all the incoming and outgoing piece and request messages with timestamp.

Documentation: (a) Explain briefly the design modification needed. (b) Draw and explain the program flowcharts. (c) Draw an explain the new message sequences between the seeder and the leechers.

Experiment: To be done in GENI. (a) Start two instances of TiniTorrent on two hosts. Evenly seed them (seed odd pieces in one and even pieces in the other). Let them now exchange the pieces. Trace the piece receipt time and plot it (y-axis) against piece indices (x-axis) (Fig-4.1). Describe the plot. Note the swarm completion time (b) Repeat the experiment with more (3-8) peer instances. Compare the swarm completion time (x-axis) vs. the number of peers (y-axis) in the swarm. Explain the plot (Fig-4.2). (c) Now use a swarm of size 4 in GENI and repeat the experiment where the number of pieces is varied from 128, 64, 32 and 16. Plot the swarm completion time with respect to number of pieces (x-axis) (Fig-4.3). Explain the graph. (d) Obtain the piece exchange trace with a swarm of three. Explain the trace.

SUBMISSION:

The submission is two parts- the Report and Code Distribution.

The Report: A detailed good report with all the following items is critical for good grade. Create a documentation describing (a) the architecture of your software. (b) Explain critical data structures used. (c) Explain the user interface. (d) Explain each of the experiments- the input data, objective of the experiment, locations of the peer-node in GENI, etc. (e) Then include all the graphs as prescribed.

Code Distribution: Create a README file with clear description of user interface (how to install, start, stop, use, see collected data, etc) for your program. Grader will use it to install, run and grade. In the PROJECT-REPORT.doc file answer all the documentation and experiment questions of the assignments.

Now clean up the TINITORRENT/* directories. Make sure in SOURCE you have all the C and header files, makefile, and the README file. Remove all binaries before zipping. In the DATA001 you should keep the sample *.torrent and *.tracker, *.init (and any other configuration files), and the uploaded/downloaded files. The top level TINITORRENT directory should contain your PROJECT-REPORT.doc file and an HTML file "INDEX.HTML". It should have pointers to the "README.txt" file, PROJECT-REPORT.doc, all source code and header files, and all configuration files.

Finally zip the entire TINITORRENT directory and email it to grading account at _____ with key word “TiniTorrentONE” in the subject header before the midnight of deadline date _____. For your record save a copy of the zip file with timestamp unchanged.

Email the project report to be with subject field “P2Pyou should also provide a hardcopy of the PROJECT-REPORT.doc in the next class.