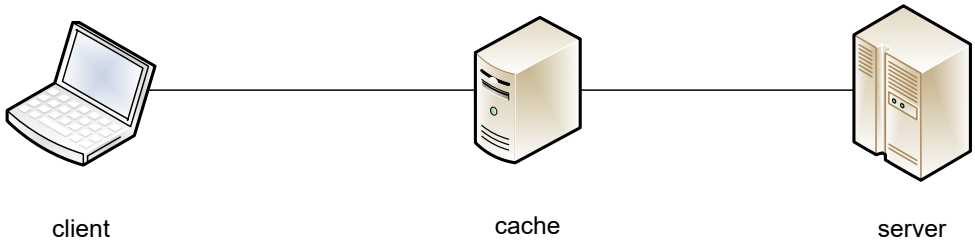
COMPSCI 711 Assignment A1

Due date: 23:00 11th April

The purpose of this assignment is to understand the techniques used by the CDN in reducing the amount of data transmission and practice developing distributed applications.

The system consists of three components as shown in the figure below.



The server provides a service that allows users to browse and download files. The cache caches the data previously downloaded by the client. It should be assumed that the three components reside at different locations. The client program interacts with the server through the cache. That is, when the client wants to browse the list of files available on the server or to download a file from the server, the client's requests are sent to the cache first and the cache forwards the requests to the server. Similarly, the server also sends the replies to the clients through the cache.
Your implementation must be based on C# and use .NET 6 Core framework (available on lab machines).

- If you use your own machine for the assignment, you should install .NET 6 SDK (available at https://dotnet.microsoft.com/en-us/download/dotnet/6.0).
- Visual Studio is an IDE for C#. The community edition is free and can be downloaded at https://visualstudio.microsoft.com/downloads/ .

There are two options for this assignment. They differ in their difficulty. Option 1 is relatively easy. The maximum mark for option 1 is 9 while the maximum mark for option 2 is 15. **You should only do and submit ONE of the options.**

Option 1 (9 marks)

Programming (6.5 marks)

In this part, you are required to implement the client program, the cache and the server. The requirements for the three programs are as below:

- The client program should provide a GUI interface that (a) allows the user to display the list of files that are available on the server, (b) allows the user to select a file from the list of available files to download, and (c) allows the user to display the contents of the downloaded file. In this option, you can assume the files to be downloaded are text files.
- The cache should cache the files previously downloaded by the client.
    - If the user requests for a file that has the same name as a previously downloaded file, the cached copy of the file should be returned to the user.
    - The cache should keep a log to record of the activities of the cache. For each user's file downloading request, the log should indicate whether the request is satisfied by a cached file, or the requested file needs to be downloaded from the server. For example, when a user's request is received, the following log entry might be created:
      user request: file xyz at 10:27:00 2022-03-01
      response: cached file xyz
      or
      user request: file xyz at 10:27:00 2022-03-01
      response: file xyz downloaded from the server
    - The cache should provide a GUI interface that allows the user to view the contents of the cache's log and the list of files that are cached on the server.
    - The interface should provide a function (e.g., a clear button) that allows the files in the cache to be cleared (i.e., deleted).
    - It should be assumed that the cache has sufficient space to hold all the downloaded files. That is, you do not need to consider cache replacement policy.
- The server program should provide (a) an operation that allows the user to download a file with a given name, and (b) an operation that lists the names of the files available on the server.

Report (2.5 marks)

Write a report about your implementation. Your report should include the following.

(a) Which option you have implemented.
(b) Clearly indicate whether you have tested your programs on a lab machine.
(c) Instructions on how to run your programs.
(d) Consider the techniques that you need to use if you have chosen to implement option 2. Compare and contrast the techniques used in option 1 and 2 in terms of the response time perceived by the user, the amount of network bandwidth that can be saved, and the amount of computation being carried out by the cache and the origin server.

Save your report as a PDF file. Name it as *report.pdf*.

Option 2 (15 marks)

Programming (12 marks)

In this part, you are required to implement the client program, the cache and the server. The cache and the server should allow fragments of the files to be cached. In your implementation, the fragmentation of the files should be carried out automatically. That is, the users do not need to rewrite the files for caching purpose. The files to be downloaded are image files.

- The client program should provide a GUI interface that (a) allows the user to display the list of files that are available on the server, (b) allows the user to select a file from the list of available files to download, and (c) allows the user to display the contents of the downloaded file (i.e., display the downloaded image).

The requirements for the cache are as below:

- When a user requests for a file, if there are some cached data that can be used to construct the requested file, the cached data should be used, and only the data that do not exist on the cache should be downloaded from the server.
- The cache should keep a log to record of the activities of the cache. For each user's file downloading request, the log should indicate the percentage of the file that is constructed using the data cached on the server. The percentage is defined as "the size of the cached data used to construct the file / the size of the file" where the size is measured in bytes. For example, when a user's request is received, the following log entry might be created:
  user request: file xyz at 10:27:00 2022-03-01
  response: 82% of file xyz was constructed with the cached data
- The cache should provide a GUI interface that allows the user to view the contents of the cache's log, the list of cached data fragments and the contents of the selected data fragment cached on the cache server. The contents of the data should be displayed as a sequence of bytes in hexadecimal numbers. Each byte consists of two hexadecimal digits. For example, the contents of a data segment might be "1F02…" where "…" denotes more hexadecimal digits. Your program must display the exact contents. That is, you cannot display "…" as output.
- The size of each data fragment should be around 2KB. Otherwise, your programs might not work properly for the test cases used by the marker.
- The interface should provide a function (e.g., a clear button) that allows the files in the cache to be cleared (i.e., deleted).
- It should be assumed that the cache has sufficient space.

The server program should provide the following:

- An operation that allows the user to download a file with a given name.
- An operation that lists the names of the files available on the server.
- Operations that are necessary to allow the cache to download fragments of the files.

The mark that you gain depends on whether the scheme that you use can maximize the reusable data on the cache. Two example files that differ in 3 bytes are given as an example. You should test your programs thoroughly with more test data.

Report (3 marks)

Write a report about your implementation. Your report should include the following.

(a) Which option you have implemented.
(b) Clearly indicate whether you have tested your programs on a lab machine.
(c) Instructions on how to run your programs.
(d) Describe the techniques that you use to determine which portions of a file need to be downloaded from the server.
(e) Consider the techniques that you need to use if you have chosen to implement option 1. Compare and contrast the techniques used in option 1 and 2 in terms of the response time perceived by the user, the amount of network bandwidth that can be saved, and the amount of computation being carried out by the cache and the origin server.

Save your report as a PDF file. Name it as *report.pdf*.

Submission

You should provide a batch file to start your programs.
Pack your implementations, batch file and report into file A1.zip. Submit A1.zip through Canvas.
NO email submission will be accepted.