

Project 01 [25 points]**Deadline November ???th**

This project has the aim to stimulate students to develop a prototype for a distributed application. In this activity we mainly focus on the development of the business logic and data access and storage, addressing the intrinsic problems and challenges related with the distributed nature of the applications. The implementation have to meet all the requirements and take profit of the technology used to provide the most optimal solution possible working without both execution and security errors. It is not important to provide a nice Client interface. You only have to implement a functional implementation that allows testing the operations of the distributed application.

The practice consists in implementing a node to generate a P2P network for sharing files.

1st level Functions [50%]

- The implementation will consist of a node, that can be the client of one other node and the server of many nodes. In this way, the network will be created.
- On startup, the node will indicate the IP and PORT of their connecting node (server), or it can start a new network by starting without any linked node, creating an isolated node.
- Each node will provide the contents of the files inside a custom folder (text, music or video) to the network, making them available for the rest of nodes, this files can also have a title, keywords and description. The user will be able to set these information once the server has started and the files recognized. If the information has not been set, the Name will be the filename and the description and keywords will be empty.
- To identify duplicated files, the nodes will also apply a hash to the file to generate a unique identifier of the file. The hash operation will be the same for all nodes, so same hash will mean that the file is the same, even if the file has different filename and is in a different node.
- Any node in the network will be able to list the different files in the network. If two files have different name/keywords/description, but referencing the same file(hash), it will show both information related to the same file.

Example:

List:

- Hash: fdsfsafdafdegecesagegcdeasfd
 - Name: [theWire.mp4, theWire_s01e01.avi]
 - Description:[first episode of the wire, first episode of the wire HBO tv show]
 - Keywords: [tv show, thriller, action]
-
- The Search must be done, by requesting all contents of all nodes in the network, thus we have to take care in order to avoid requesting loops due to connection typology, or requesting the same node multiple times.
 - The Node can download a digital content using its key from the Node that has the content. The file must be transferred physically to the requesting node folder.
 - The client can modify the title or delete the contents only in their own folder.

2on level Functions [30%]

- The Nodes can search the contents doing a partial search on titles and descriptions and keywords. This search must be applied in the whole network.

(10%)

- The downloading of the document is parallelized by splitting the file in chunks of fixed size (1MB), the requesting node will ask the different nodes that have the same file for one chunk in a parallel way, distributing the downloading between all nodes that have the same file. And Joining the different parts into the main file once all parts are complete to rebuild the file.(20%)

3rd level Functions [10 %]

- To avoid saturating the Node, the user can configure how many uploading and downloading threads can be active at a time. By setting this value, once the server has enough threads working in the task, the following requests will be placed in a queue, and served when some thread is finished.

Quality Features [10%]

The next features provides extra points.

- Alternatively you can use a relational database or NoSQL database
- Data storage is done considering optimality
- Data communications are done considering optimality

Considerations

Considerations

You face the development of a distributed application. Your solution should provide correct implementation of the functionalities and consider efficiency, fault tolerance, performance and basic security issues. You should properly answer questions such as:

- How to store physically the digital contents?
- How to reduce the number of transfers of digital contents?
- How to implement an efficient search?
- It is important to provide security over the contents.
- It is important to provide transparency about the content location.
- Is not necessary to implement a graphical interface.
- To show the execution of your distributed application, you must execute the application in a distributed environment with different servers and clients. The implementation should consider this situation.
- You can choose the technology to develop the project, but all decisions must be justified. The technology you choose must consider facilities to efficiently develop distributed applications with access to remote objects or services, etc. The technology chosen cannot be an excuse neither to satisfy the project requirements nor for a correct implementation of the functionalities.

Deliveries

Report content

Create a report with the following contents:

1. Provide the UML class diagram. Describe for each class the main functionality, methods, data structures, etc..
2. Summarize the main design decisions done in this project such as class hierarchy, data structures, etc., and justify them.
3. Describe the implemented functionalities and provide the sequence diagrams for the functions: upload a content, execute a search and download a content in a remote node.
4. Run two several nodes. Describe and explain the outcome.
5. Run the nodes in different hosts. Describe and explain the outcome.
6. Execute the main use cases. Describe and explain the outcome.

Instructions

Work in pairs to develop the distributed application project. Submit a report with the contents specified above. The final source code should be published in a repository where the professors have access. Do not forget to indicate in the report the time spent on the activity. **There will be a face-to-face presentation with the professors. In the case a group deliver the project after the deadlines there will be penalizations on the score.**