LULEÅ TEKNISKA UNIVERSITET

Mini-project **D7001D Network programming and distributed applications** (LP1 2022)

Mini-project: A special kind of publish subscribe system

LYCKA TILL!

For the home exam you are requested to apply all the skills and technologies you mastered during the course and realize a mini project as per the following description.

You will design and implement a distributed networking system implementing a special kind of publish-subscribe functionality.

The scenario you are asked to implement assumes a need for two or more clients to communicate anonymously over a public publish-subscribe system. That is the network should not know the true destination of the particular data chunk nor the true origin of the data chunk.

For this purpose a client, that originates data will deploy a special kind of embedding of the destination id into the data chunk itself.

The encoded data is then sent to a publish-subscribe core, which stores the encoded data chunks based on a similarity measure.

The client-recipient of the data knows the ID of the data chunks in way not covered by this project. The recipient communicates this ID to retrieve the stored encoded data from the publish subscribe system.

The publish-subscribe system should be scalable to accommodate dynamically varying population of clients.

Below you find some details of the technical specification of the system functionality.

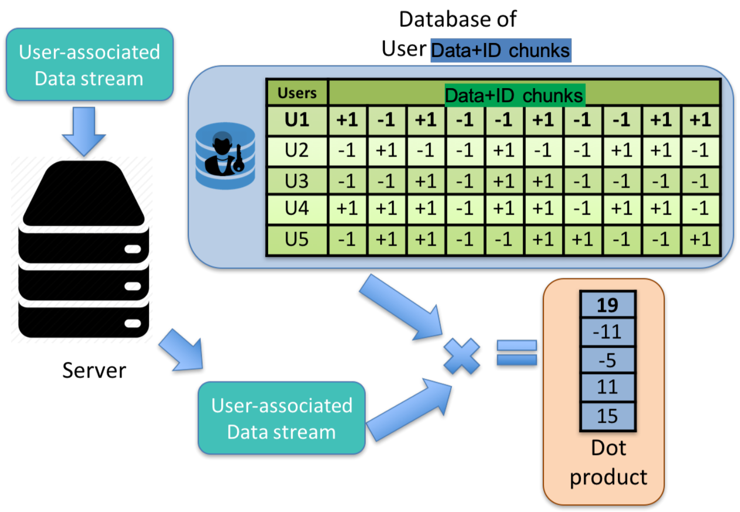
Client- data originator

- For a given volume of data a client randomly generates a vector of -1 and +1 of dimensionality D (the test should be performed with dimensionality 1000)
- The generated ID vector is published on the client's webpage
- The ID vector is embedded into data according to the following routine:
    1. The data is converted to a binary vector $\{0,1\}_b$
    2. The binary vector is converted to a bipolar vector{-1,1}
    3. The ID vector is first multiplied element-wise with the bipolar data vector and then element-wise added to the product. That is: Data*=(Data x ID) + ID.
    4. After the addition the result (Data*) will have values {-2, -1,0,1,2} in each element.
- The data with an embedded ID (Data*) is sent to the known IP address of the publish-subscribe system

Client- data recipient

- The client data recipient knows the ID of the user it wants to receive
- It request all data with this label from the server.
- When receiving the data Data* chunk the receiver perform inverse operations to step 3 above in order to retrieve the data.

Server side. Search for an ID and publishing the data with embedded ID. The functionality of the server is depicted in the figure below.



Users	Data+ID chunks									
U1	**+1**	**-1**	**+1**	**-1**	**-1**	**+1**	**-1**	**-1**	**+1**	**+1**
U2	-1	+1	-1	-1	+1	-1	-1	+1	+1	-1
U3	-1	-1	+1	-1	+1	+1	-1	-1	-1	-1
U4	+1	+1	+1	-1	+1	+1	-1	+1	+1	-1
U5	-1	+1	+1	-1	-1	+1	+1	-1	-1	+1

- The server does not know the ID of the clients. What it sees is only the data with the embedded IDs
- It maintains a so-called autoassociative table. This is a memory which is addressed by the data itself.
- Initially the table is empty.
- It also maintains N buffers in memory for storing data of N users.
- If the table is empty, the received Data* chunk is inserted and is associated with a pointer to a place in memory which will keep data for this user. It also saves the received chunk there.
- If table is not empty, the server performs and associative search:
    1. It computes a dot product of the received Data* chunk with all the entries in the table.
    2. If the result is larger than a threshold (a parameter). It selects the entry with the maximum value of the dot product and stores the received chunk in the buffer linked to this entry.

That's it. Now here are the criteria for grading the exam:

1. To get the HIGHEST grade – 5: well documented code, fully implemented functionality, user-friendly intuitive GUI + clean *.PPT presentation describing the design of the system and performance evaluation
2. To get 4: working implementation of the core functionality, but with hard coded parameters and basic GUI, satisfactory performed tests and the PPT presentation of the design
3. To get 3: partially working solution implementing the key networking functionality, PPT presentation describing the design and encountered difficulties and the ways to overcome them.

GOOG LUCK!