TDDD25: Distributed Systems
Programming Project

Petru Eles Ivan Ukhov

Computer and Information Science
Linköping University

January 25, 2016

Contacts

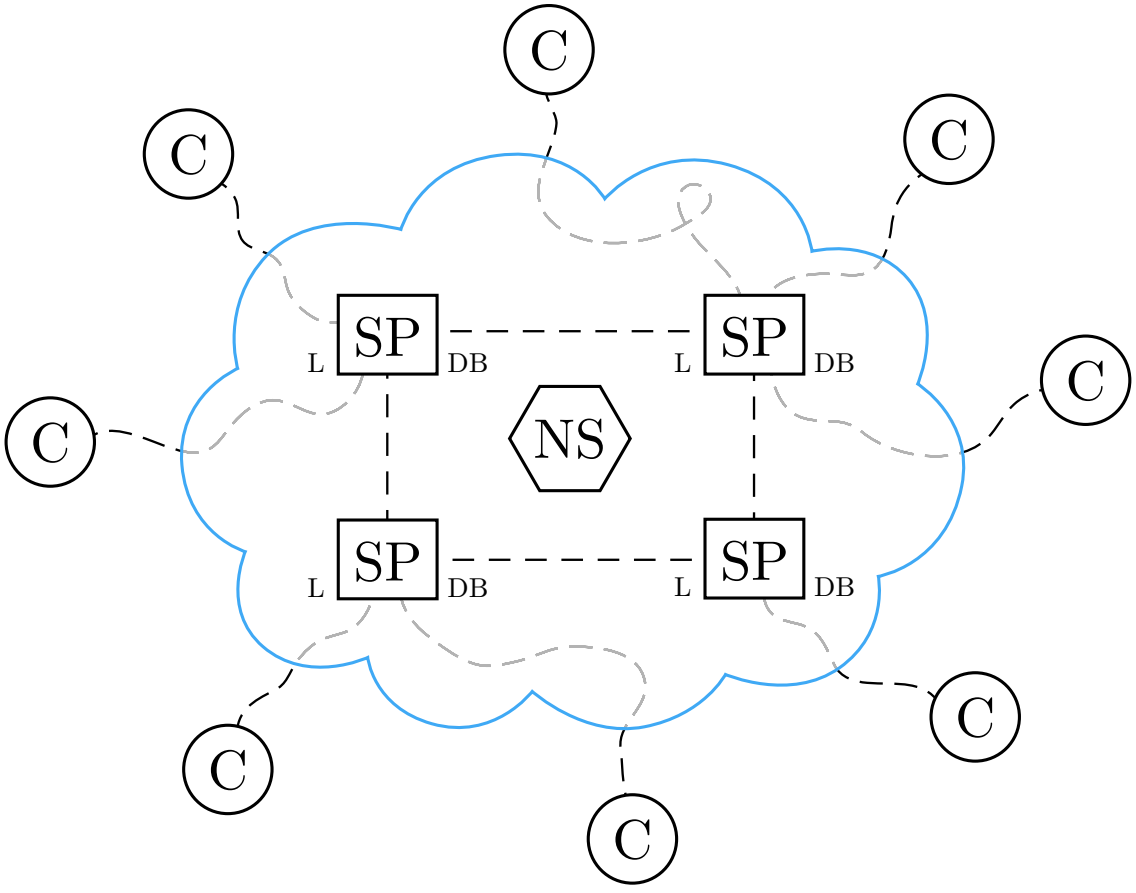
Ivan Ukhov

ivan.ukhov@liu.se
Office 329:228, Building B

Organization

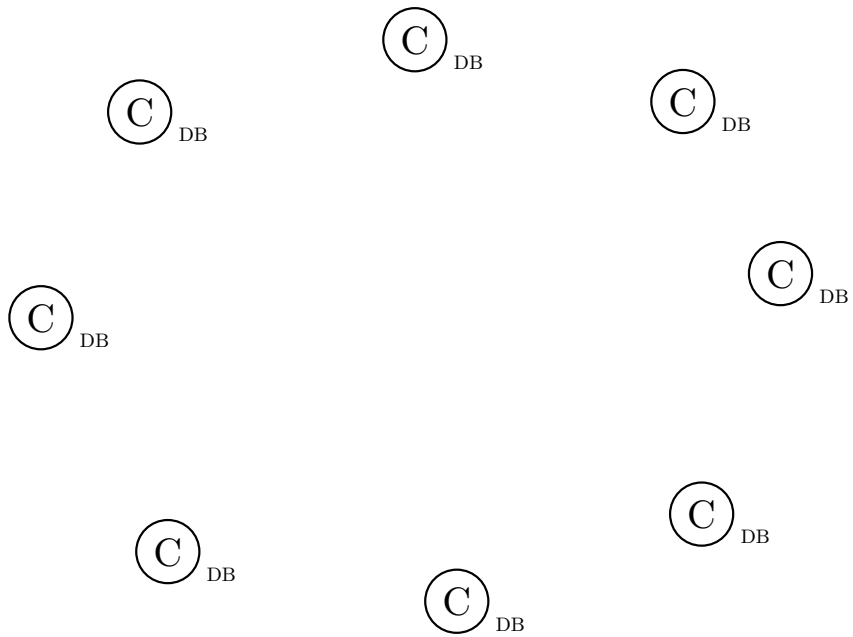
- 2 groups
- 7 sessions for each group
- $1 + 5$ assignments
- Registration deadline: January 31
- Completion deadline: two weeks after the exam

Distributed Database



C — client
SP — server/peer
NS — name service
DB — database
L — lock

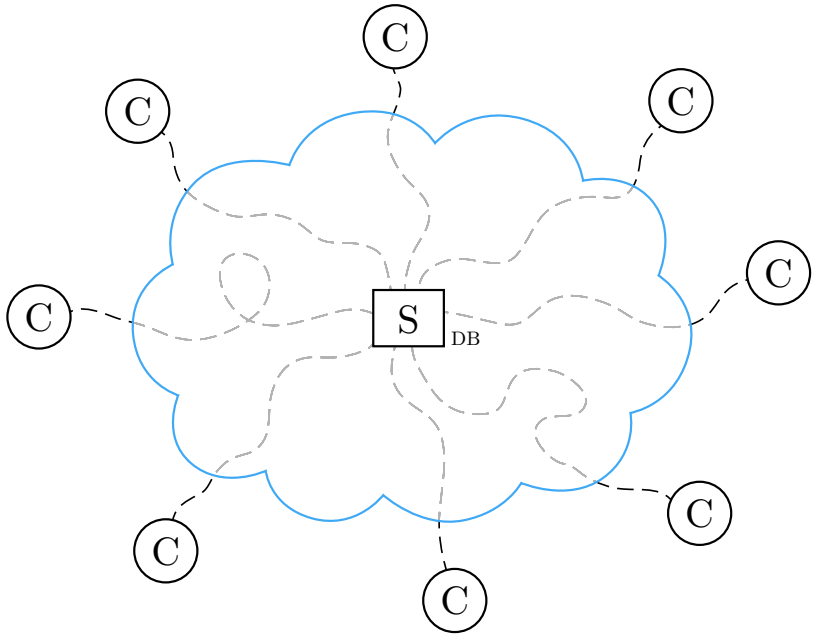
Assignment 0: Standalone Database



- Local database for each client
- TODO: complete the implementation the read and write operations of the database

`https://gitlab.ida.liu.se/tddd25/labs/raw/master/doc/assignment0.pdf`

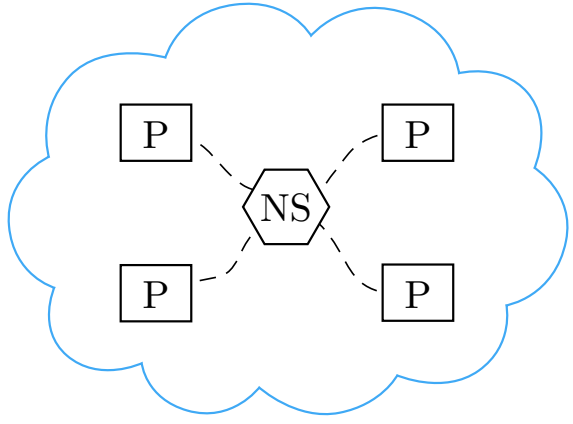
Assignment 1: Client-Server Database



- Centralized database
- TODO: implement the client/server communication mechanism ensuring thread-safeness

`https://gitlab.ida.liu.se/tddd25/labs/raw/master/doc/assignment1.pdf`

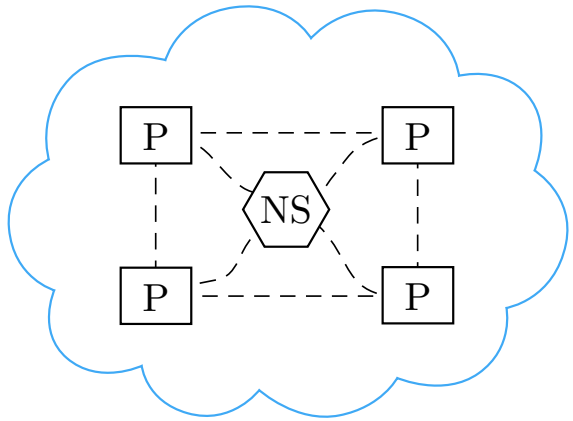
Assignment 2: Object Request Broker



- Name service and object request broker (ORB)
- Abstract away the communication part from functionality
- TODO: complete the implementation of the ORB

`https://gitlab.ida.liu.se/tddd25/labs/raw/master/doc/assignment2.pdf`

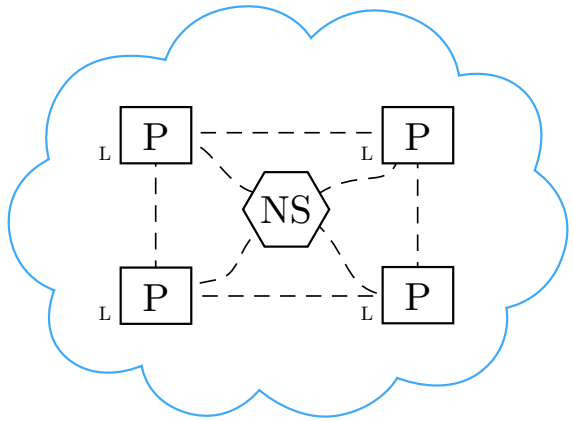
Assignment 3: Peer-to-Peer Communication



- Smart mechanism for keeping track of peers
- TODO: complete the routines dealing with the peers who are joining the system and those who are leaving

`https://gitlab.ida.liu.se/tddd25/labs/raw/master/doc/assignment3.pdf`

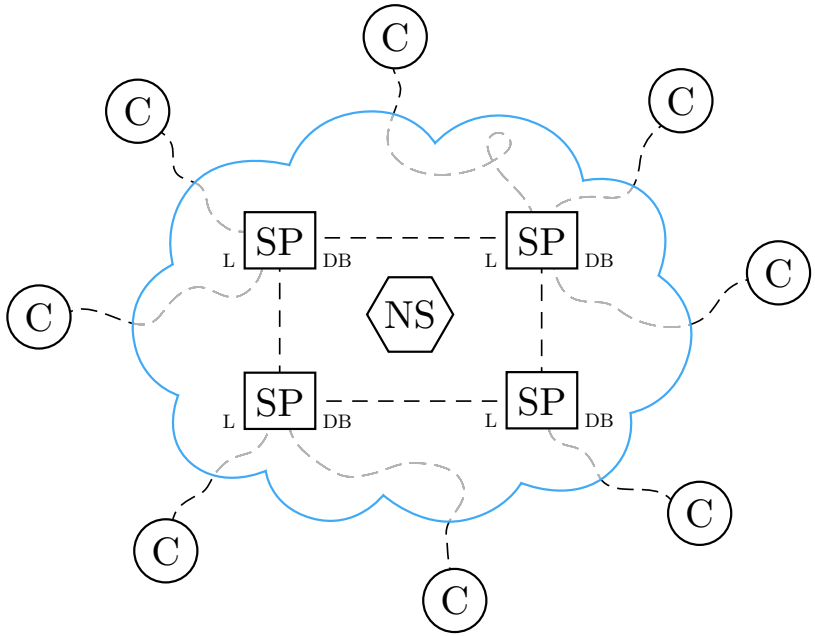
Assignment 4: Distributed Locks



- Distributed mutual exclusion to control concurrent operations
- TODO: implement the second Ricart–Agrawala algorithm

`https://gitlab.ida.liu.se/tddd25/labs/raw/master/doc/assignment4.pdf`

Assignment 5: Client-Server Database with Replicas



- Everything together
- TODO: complete the implementation of the server/peer using all the previously developed components

`https://gitlab.ida.liu.se/tddd25/labs/raw/master/doc/assignment5.pdf`

Implementation

- Multi-threaded object-oriented code in Python 3
- Communication via objects serialized in JSON
- Data transfer through TCP sockets

Repository Structure

- doc/
- src/
  - lab0/
  - lab1/
  - lab2/
  - lab3/
  - lab4/
  - lab5/
  - modules/
    - Common/
    - Server/

Submission

- No written reports are needed
- Demonstrate your solutions in class
- Email modified files

Good luck!