

TDDD25: Distributed Systems Programming Project

Petru Eles Ivan Ukhov

Computer and Information Science
Linköping University

January 25, 2016

Contacts

Ivan Ukhov

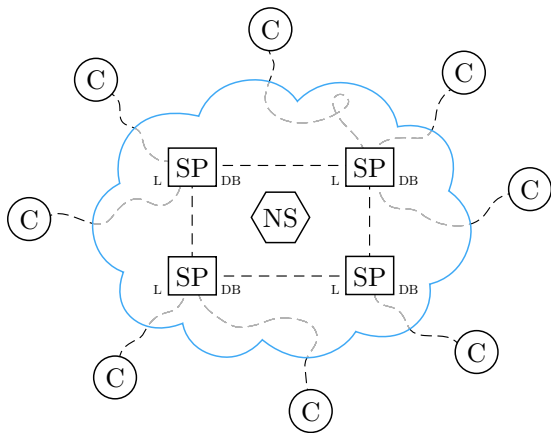
ivan.ukhov@liu.se

Office 329:228, Building B

Organization

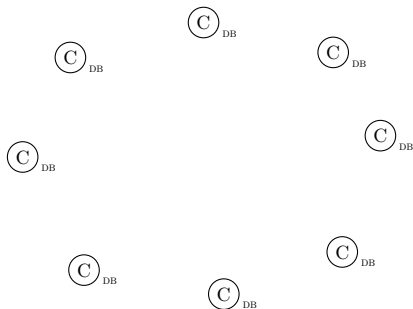
- 1 teaching session
- 7 lab sessions
- 1 + 5 labs
- 2 groups
- Registration deadline: January 31
- Completion deadline: two weeks after the exam

Distributed Database



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

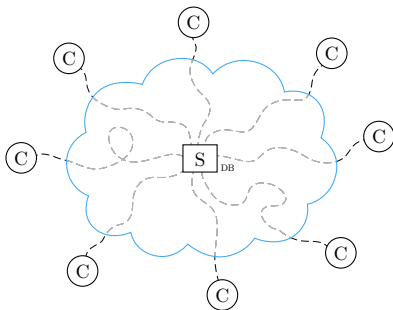
Lab 0: Standalone Database



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

<https://gitlab.ida.liu.se/tddd25/labs/raw/master/doc/lab0.pdf>

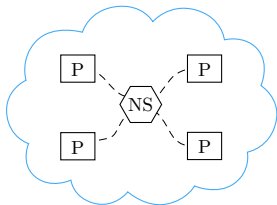
Lab 1: Client-Server Database



- Centralized database
- **TODO**: complete the implementation of the client/server communication mechanism

<https://gitlab.ida.liu.se/tddd25/labs/raw/master/doc/lab1.pdf>

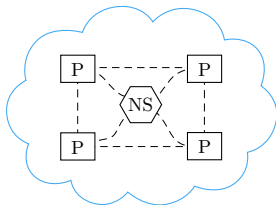
Lab 2: Object Request Broker



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

<https://gitlab.ida.liu.se/tddd25/labs/raw/master/doc/lab2.pdf>

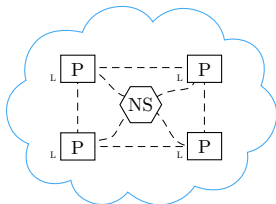
Lab 3: Peer-to-Peer Communication



- Smart mechanism for keeping track of peers
- **TODO**: complete the functionality dealing with the peers joining or leaving the system

<https://gitlab.ida.liu.se/tddd25/labs/raw/master/doc/lab3.pdf>

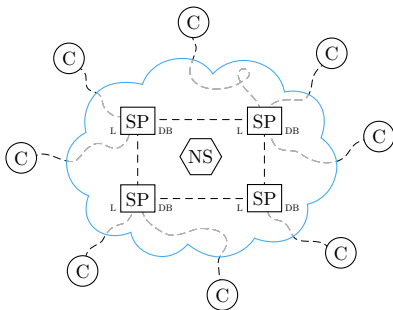
Lab 4: Distributed Locks



- Distributed mutual exclusion to control concurrent operations
- **TODO**: complete the implementation of the second Ricart–Agrawala algorithm

<https://gitlab.ida.liu.se/tddd25/labs/raw/master/doc/lab4.pdf>

Lab 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/lab5.pdf>

Implementation

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

Repository

- 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!