TDDD25 – Distributed Systems Lesson

Petru Eles, Sergiu Rafiliu, Ivan Ukhov

Department of Computer and Information Science,
Linköping University, Sweden

January 17, 2013

Lab Organization

Laboratory Assistant:	**Sergiu Rafiliu**
Groups A, B	email: sergiu.rafiliu@liu.se
	office: B building, 329:232
Laboratory Assistant:	**Ivan Ukhov**
Groups C, D	email: ivan.ukhov@liu.se
	office: B building, 329:228

Lab Organization

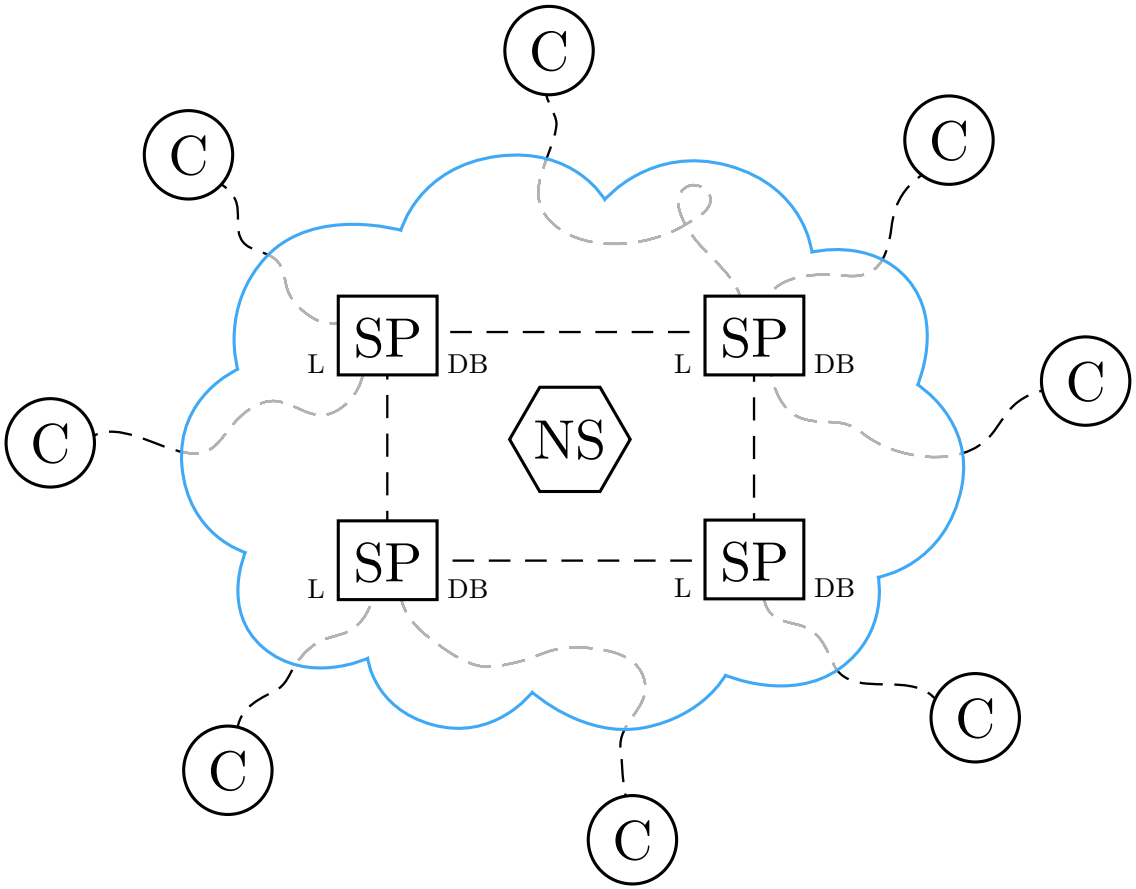
- 4 lab groups
- 14 hours/student (supervised)
- 7 lab sessions
- 5 lab assignments (+ lab 0)
- 2 points
- Home page

  `http://www.ida.liu.se/~TDDD25`
- Lab registration

  `http://www.ida.liu.se/webreg`
- Deadlines:
  - lab sign-up: 31 January
  - hand-in the assignments: 2 weeks after the exam.

Lab Assignments

A Replicated Client-Server Database System

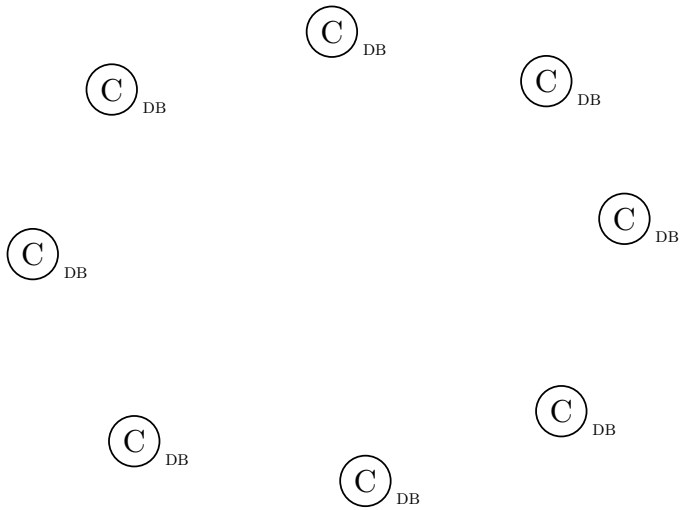


C – client
SP – server/peer
NS – name service
DB – database
L – lock

Lab Assignments (cont'd)

- object-oriented code, written in Python
  - tutorial:
    `http://docs.python.org/release/2.7.3/tutorial/index.html`
  - library reference:
    `http://docs.python.org/release/2.7.3/library/index.html`
- communication through TCP using `sockets`
- python objects serialized for communication into `JSON` format.
- the code is multi- `threaded`.

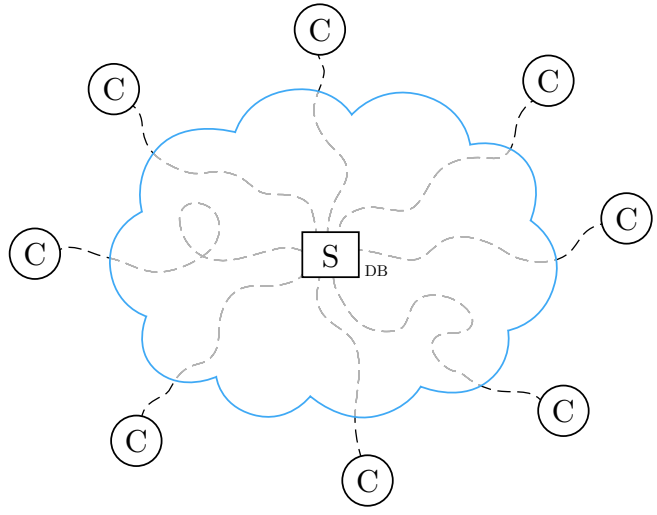
Assignment 0 – Standalone Database



- Warm-up assignment.
- The database is connected directly to the client.
- TO DO: complete the implementation of the database library with read and write operations.
- More details at:

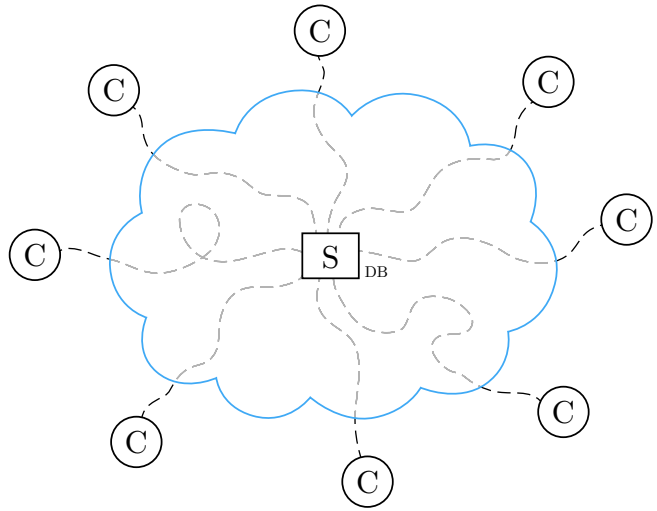
  `http://www.ida.liu.se/~TDDD25/labs/assignment0.pdf`

Assignment 1 – Client-Server Database System



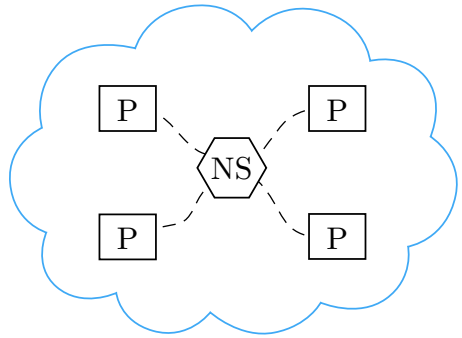
- All clients are connected to one single server that holds the database.
- TO DO:
  - implement the communication part in the client and the server.
  - implement mutual exclusion in the server.

Assignment 1 – Client-Server Database System (cont'd)



- Hint: check out the global function `type()` and the object attributes `__class__` and `__name__`.
- More details at:
  `http://www.ida.liu.se/~TDDD25/labs/assignment1.pdf`

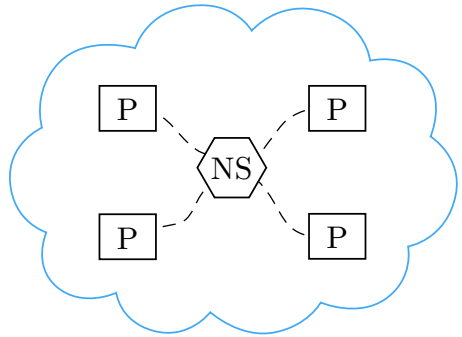
Assignment 2 – Middleware: Object Request Broker



- Abstract away the (messy) communication from the functionality of a distributed system
- Object Request Broker library provides:
  - *Stubs* – makes the server seem a local object for the client
  - *Scheleton* – makes the clients seem as local objects (running in parallel threads) for the server.
  - registration with the name service.

Assignment 2 – Middleware: Object Request Broker

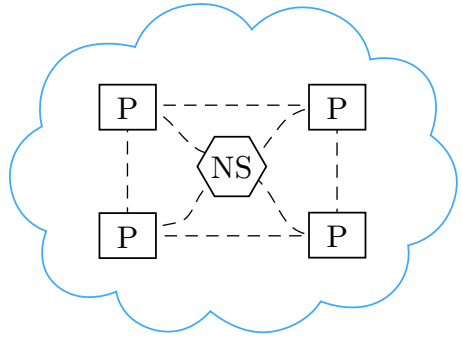
(cont'd)



- TO DO: complete the implementation of Stub and Scheleton
- More details at:

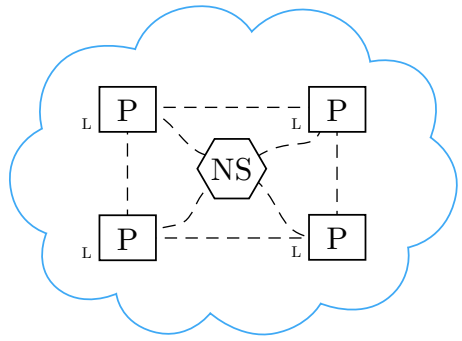
  `http://www.ida.liu.se/~TDDD25/labs/assignment2.pdf`

Assignment 3 – Middleware: Peer-to-Peer Communication



- Implement a simple form of *global state* that enables peer-to-peer communication among the *servers* that hold *database* replicas. The *state* is simply a list of peers.
- TO DO: complete the necessary routines when a new peer joins and leaves the system.
- More details at:
  `http://www.ida.liu.se/~TDDD25/labs/assignment3.pdf`

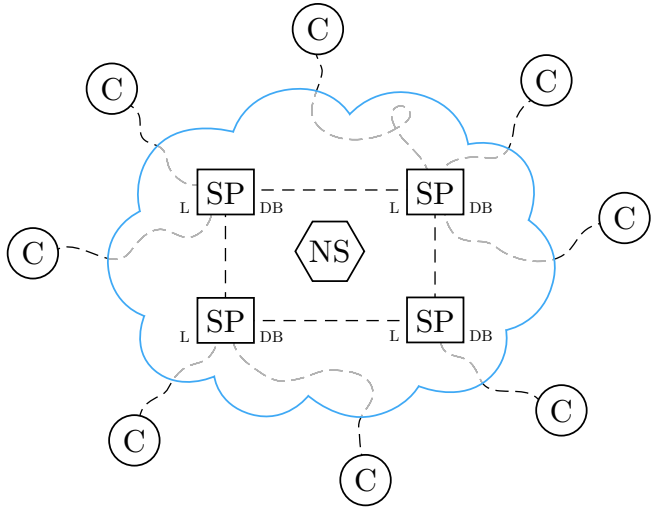
Assignment 4 – Middleware: Distributed Locks



- In order to synchronize database replicas, one server/peer needs to be able to temporarily block all other servers from writing to their own replica. Here mutual exclusion distributed among all servers/peers is needed.
- TO DO: complete the implementation for distributed mutual exclusion using the *second Rikard-Agravara algorithm*.
- More details at:

  `http://www.ida.liu.se/~TDDD25/labs/assignment4.pdf`

Assignment 5 – Replicated Client-Server Database System



- Put everything together.
- TO DO: complete the server implementation using all the previously developed middleware.
- More details at:

  `http://www.ida.liu.se/~TDDD25/labs/assignment5.pdf`

Good Luck