

Prof. Dr. Nikolaus Augsten

Jakob-Haringer-Str. 2  
5020 Salzburg, Austria  
Phone: +43 662 8044 6347  
E-Mail: nikolaus.augsten@plus.ac.at



PARIS  
LODRON  
UNIVERSITÄT  
SALZBURG

---

Parallel and Distributed Data Management  
Summer Semester 2025

Midterm II  
24.06.2025

---

Name: \_\_\_\_\_ Student ID: \_\_\_\_\_

---

Hints

---

- Check whether you received all pages of the exam (5 pages).
- Write your name or your student ID on each sheet of the exam and hand in all pages.
- All answers are expected to be written on the exam sheets.
- Clearly highlight and enumerate additional pages that are used for longer answers. Match your text with the according exercise.
- Only use pencils that are permanent and non-red colored.
- Use the notation and techniques discussed in the lecture.
- Exercises with more than one solution are not graded.
- You are allowed to use one A4 sheet with your personal notes (single-sided, hand written or printed).
- Exam duration: 60 minutes

Signature \_\_\_\_\_

---

Grading

Filled by the examiner

---

Exercise	1	2	3	4	Sum
Total points	3	4	4	4	15
Points reached					

---

**Exercise 1 - Commit Protocols.****3 Points**

---

Mark the following statements as true (**T**) or false (**F**).

**Wrong answers result in point deductions!**

- (1) 2-Phase Commit does not block as long as the coordinator is reachable.
- (2) In 2-Phase Commit, when a site recovers (due to a failure) and finds a log entry `<ready T>` (but no `<commit T>` or `<abort T>`), then T had failed.
- (3) The first phase in 2-Phase Commit is identical to the first phase in 3-Phase Commit.
- (4) 3-Phase Commit avoids the blocking problem under network partitioning.
- (5) In 3-Phase Commit, when a site recovers (due to a failure) and finds a log entry `<abort T>`, no action is required.
- (6) The persistent messaging protocol avoids the receiver relation *received\_messages* to grow too large by removing all messages that are older than a user-defined timeout.

## Exercise 2 - Persistent Messaging.

4 Points

Consider a sender  $S$  that sends a message to receiver  $R$  using the persistent messaging protocol. Table 1 shows the initial entries in the relations  $messages\_to\_send$  of the sender and  $received\_messages$  of the receiver. Newer events have larger time stamps.

$messages\_to\_send$				$received\_messages$			
number	message	time	ack	number	message	time	ack
1	$Q \leftarrow Q + 9$	2	received	7	$Q \leftarrow Q + 3$	5	sent
3	$A \leftarrow A + 3$	3	received	8	$B \leftarrow B - 9$	7	sent
7	$Q \leftarrow Q + 3$	5		9	$C \leftarrow C - 6$	8	sent
8	$B \leftarrow B - 9$	7	received				
9	$C \leftarrow C - 6$	8					

Table 1: Relations  $messages\_to\_send$  at sender  $S$  and  $received\_messages$  at receiver  $R$ .

- (1) Assume that  $S$  receives the acknowledgement for the message with number 9. Compute the value of  $T_{OLD}$ .
- (2) Show the relation  $received\_messages$  after receiver  $R$  has received and processed the value of  $T_{OLD}$ .

## Exercise 3 - Vector Clocks.

4 Points

Assume a single data item  $Q$  that is replicated on sites  $S_1$ ,  $S_2$ , and  $S_3$ .

A site  $S_i$  can do (i) a local write on  $Q$ ,  $W(Q)$ , which changes the value of the local copy  $Q_i$ , or (ii) copy the value from a different site  $S_j$ ,  $C(S_j)$ ,  $j \neq i$ , which copies the value of  $Q_j$  to  $Q_i$ . Show the vector clocks resulting from the schedule in Figure 1 and indicate for each pair of local copies  $(Q_1, Q_2)$ ,  $(Q_1, Q_3)$ , and  $(Q_2, Q_3)$  whether their values stem from a branching history. All vectors are initialized with the zero vector.

*Note: Local reads, which will typically precede a local write in a real schedule, are not relevant for conflict detection and omitted from the schedule.*

$S_1$	$S_2$	$S_3$
$W(Q)$		
	$C(Q_1)$	
	$W(Q)$	
		$C(Q_1)$
		$W(Q)$
$C(Q_3)$		
$W(Q)$		

Figure 1: Schedule on replicated data item  $Q$ .

---

Exercise 4 - *Consistency*.4 Points

---

Consider quorum consensus replication. Quorum consensus requires two conditions for its correctness:  $Q_r + Q_w > S$  and  $2 \cdot Q_w > S$ . Consider the following relation distributed across three sites  $S_1$ ,  $S_2$ , and  $S_3$  with weights 6, 2, and 2, respectively. Show a schedule of reads and writes that violates strong consistency if the quorum consensus conditions are **not fulfilled**. Initially, all sites store identical copies of the relation, and all version numbers on all sites are zero.

<i>AccountBalance</i> on $S_1$ , $S_2$ , and $S_3$		
accountid	balance	version
1	100	0
2	300	0
3	150	0