



HS 2011

Prof. Dr. Roger Wattenhofer
B. Keller, S. Welten, R. Eidenbenz, R.Meier

Prüfung

Verteilte Systeme

Teil 2

Freitag, 10. Februar 2012
9:00 – 12:00

Die Anzahl Punkte pro Teilaufgabe steht jeweils in Klammern bei der Aufgabe. Sie dürfen die Fragen englisch oder deutsch beantworten. Begründen Sie alle Ihre Antworten und beschriften Sie Skizzen und Zeichnungen verständlich. Schreiben Sie zu Beginn Ihren Namen und Ihre Legi-Nummer in das folgende dafür vorgesehene Feld.

Name	Legi-Nr.

Punkte

Frage Nr.	Erreichte Punkte	Maximale Punkte
9		23
10		15
11		16
12		17
13		19
Total		90

9 Multiple Choice (23 Punkte)

Beurteilen sie ob die folgenden Aussagen richtig oder falsch sind und kreuzen sie die entsprechenden Felder an. Eine richtig beurteilte Aussage gibt 1 Punkt, eine nicht beurteilte Aussage 0 Punkte, eine nicht richtig beurteilte Aussage **-1 Punkt**. Die gesamte Aufgabe wird mit minimal 0 Punkten bewertet.

A) Konsensus

Aussage	wahr	falsch
Ein Ausführungsbaum (execution tree) eines korrekten Konsensus Protokolls hat immer mindestens einen kritischen Zustand.	<input type="checkbox"/>	<input type="checkbox"/>
Es gibt was die Anzahl verschickter Nachrichten betrifft keinen besseren Konsensus-Algorithmus als den King-Algorithmus.	<input type="checkbox"/>	<input type="checkbox"/>
Ein korrekter warte-freier Konsensusalgorithmus, der nur die Test&Set Primitive verwendet, kann auch ohne Test&Set, dafür aber mit doppelt so vielen Lese- und Schreib-Registern implementiert werden.	<input type="checkbox"/>	<input type="checkbox"/>
Mit Hilfe von Authentifizierung kann man binären Konsensus sogar mit beliebig vielen Byzantinischen Prozessen erreichen.	<input type="checkbox"/>	<input type="checkbox"/>
Der Shared Coin Algorithmus, der in der Vorlesung besprochen wurde, braucht im Erwartungswert $n/2$ Runden (wobei n die Anzahl teilnehmender Prozessoren ist).	<input type="checkbox"/>	<input type="checkbox"/>

B) Consistency und Commit Protokolle

Aussage	wahr	falsch
Die Anzahl gesendeter Nachrichten in PBFT ist in $O(n)$ wenn n die Anzahl der Backups ist.	<input type="checkbox"/>	<input type="checkbox"/>
Die Anzahl gesendeter Nachrichten in Zyzzyva ist in $O(n)$ wenn n die Anzahl der Backups ist.	<input type="checkbox"/>	<input type="checkbox"/>
Wenn Daten nur an einem Ort gespeichert sind (keine weiteren Replikas) ist die <i>read-your-writes-consistency</i> immer garantiert.	<input type="checkbox"/>	<input type="checkbox"/>
Paxos benötigt einen nicht-byzantinischen Koordinator, damit es korrekt ausgeführt werden kann.	<input type="checkbox"/>	<input type="checkbox"/>
Wenn eine Execution linearisierbar ist, ist damit immer eine eindeutige client partial order pro client definiert.	<input type="checkbox"/>	<input type="checkbox"/>
Sich korrekt verhaltende Knoten eines Two-phase commit Protokolls können nach Abschluss aller durch two-phase commit gegebener Kommunikation immer entscheiden, ob ein Koordinator das Protokoll korrekt ausgeführt hat (ohne mit anderen Knoten zu kommunizieren).	<input type="checkbox"/>	<input type="checkbox"/>

C) Selfish Peers & P2P

Aussage	wahr	falsch
Jedes Spiel besitzt mindestens ein mixed Nash Equilibrium.	<input type="checkbox"/>	<input type="checkbox"/>
Jedes Spiel besitzt höchstens so viele mixed Nash Equilibria wie pure Nash Equilibria.	<input type="checkbox"/>	<input type="checkbox"/>
Matching Pennies besitzt genau drei Nash Equilibria.	<input type="checkbox"/>	<input type="checkbox"/>
Im Allgemeinen ist das Problem, ein Nash Equilibrium in einem Selfish-Caching-Netzwerk zu finden, NP-hart.	<input type="checkbox"/>	<input type="checkbox"/>
Der Bau der Schnellstrasse im Braess-Paradoxon führt zu einer Verschlechterung des sozialen Optimums.	<input type="checkbox"/>	<input type="checkbox"/>
Ein neuer Peer mit ID x findet seinen Platz in einem DHT Search Tree indem er nach x sucht, und sich den Suchbereich des Peers, der verantwortlich ist für x , mit eben diesem Peer aufteilt.	<input type="checkbox"/>	<input type="checkbox"/>
Gerade in diesem Moment existiert eine Hypercube-DHT mit über 1 Million Peers.	<input type="checkbox"/>	<input type="checkbox"/>

D) Locking

Aussage	wahr	falsch
Test&Set (TAS) Locks und Test&Test&Set (TTAS) Locks sind ähnlich effizient in Systemen mit einer Uniform Memory Architektur.	<input type="checkbox"/>	<input type="checkbox"/>
MCS Locks garantieren FIFO Fairness.	<input type="checkbox"/>	<input type="checkbox"/>
Beim Backoff Lock ist es einfacher, einen Lock Request abzubrechen, als beim ALock.	<input type="checkbox"/>	<input type="checkbox"/>
Wenn ein Programm schlecht programmiert ist, ist es manchmal besser, alle bis auf einen Core abzustellen.	<input type="checkbox"/>	<input type="checkbox"/>
Anderson Queue Locks (ALock) sind gut geeignet bei der Verwendung vieler Locks.	<input type="checkbox"/>	<input type="checkbox"/>

10 Nicht ganz Konsensus (15 Punkte)

In der Vorlesung haben Sie wichtige Eigenschaften von asynchronen Konsensus Algorithmen kennengelernt. In dieser Aufgabe stehen uns nur atomare Leseregister und atomare Schreibregister zur Verfügung.

- A) (6 Punkte) Nennen Sie die 3 wichtigsten Eigenschaften welche ein korrekter Konsensus Algorithmus erfüllen muss und beschreiben Sie in jeweils einem Satz, was diese bedeuten.
- B) (9 Punkte) Entwerfen Sie drei Algorithmen (für jeweils zwei Prozessoren) die jeweils nur zwei dieser drei Eigenschaften erfüllen.

11 Selfish Peers & Auktionen (16 Punkte)

A) Beantworten Sie folgende Fragen mit 1-3 Sätzen:

- 1) (2 Punkte) Beschreiben Sie in Worten, was es für ein System bedeutet, wenn sein Price of Anarchy 1 ist?
- 2) (2 Punkte) Was versteht man unter Mechanism Design?

B) In der Vorlesung haben wir eine Auktion kennen gelernt, in der der Auktionär in einer ersten Phase je ein verhülltes Gebot pro Teilnehmer annimmt, und in einer zweiten Phase den Gewinner und den Preis auswählt, welcher der Gewinner erhält für das Cachen eines Objektes. Ein Teilnehmer garantiert dabei, das Objekt zu cachen, wenn er dafür eine Zahlung erhält, die mindestens seinem Gebot entspricht. In der Second-Lowest-Price-Auktion beispielsweise erhält derjenige Teilnehmer mit dem kleinsten Gebot den Zuschlag zu einem Preis, der dem zweitkleinsten Gebot entspricht.

- 1) (7 Punkte) Entscheiden Sie für folgenden Auktionstyp, ob er truthful ist oder nicht. Begründen Sie Ihre Antwort.
 - Der Teilnehmer mit dem tiefsten Gebot kriegt den Zuschlag zu einem Preis, der dem *dritttiefsten* Gebot entspricht.

Zur Erinnerung: Eine Auktion ist *truthful* wenn für jeden Teilnehmer i gilt, dass es niemals schlechter für i ist, den wahren Value (Caching-Kosten α minus Demand w_i) zu bieten als einen anderen Wert, und in gewissen Fällen strikt besser.

- 2) (5 Punkte) Die Second-Lowest-Price-Auktion ist truthful, wenn jeder Teilnehmer sein Gebot unabhängig von den anderen Teilnehmern macht, d.h., jeder Teilnehmer kennt nur seinen Value, und die Teilnehmer sprechen sich nicht ab. Angenommen die Teilnehmer können sich absprechen und im Falle des Auktionsgewinns Bezahlung und Caching-Kosten teilen. Gibt es für die Teilnehmer eine bessere gemeinsame Strategie, als dass alle ihren wahren Value bieten?

12 Interplanetare Brieftauben (17 Punkte)

Gegeben sei das folgende Szenario: In ferner Zukunft haben die Menschen Raumstationen auf verschiedenen Planeten eingerichtet. In jeder dieser Raumstationen ist ein Server. Der Zweck dieser Server ist es mehrere Register repliziert zu speichern. Die Kommunikation zwischen diesen Servern ist aus unerfindlichen Gründen nicht über elektromagnetische Wellen möglich sondern muss durch interplanetare Brieftauben realisiert werden. Diese Brieftauben fliegen mit einer Geschwindigkeit v_B und transportieren Pakete für das Kommunikationssystem. Die Brieftauben fliegen in jedem Moment direkt auf ihr Ziel zu. Das Kommunikationssystem kennt zwei Operationen $read_A(X)$ und $write(X, val)$ um das Register X von Replika A auszulesen bzw. um den Wert val in alle Replika zu schreiben. Clientseitig kommt ein Aufruf von $write(X, val)$ sofort zurück, wogegen ein Aufruf von $read_A(X)$ bis zur Rückkehr der Antwort-Brieftaube blockiert (Siehe Beispiel).

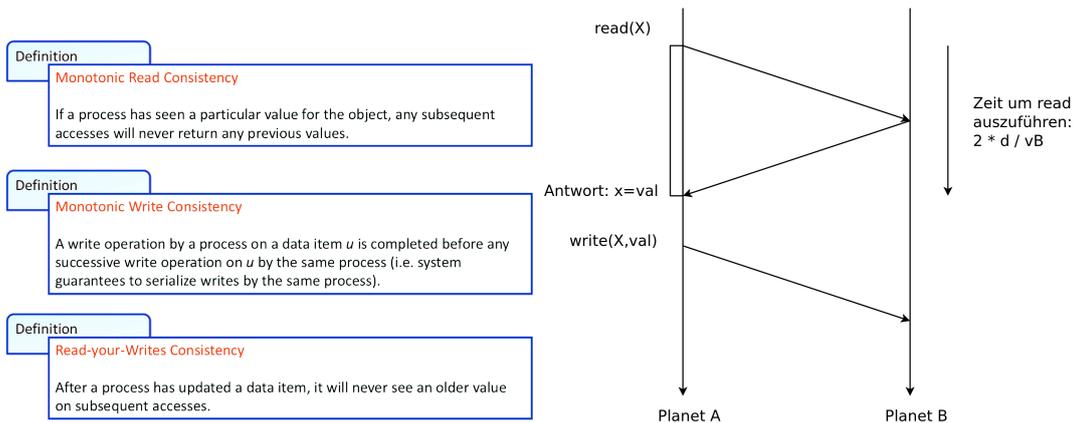


Abbildung 1: Links, die Konsistenzmodelldefinitionen aus der Vorlesung. Rechts, ein Beispiel eines Lese- und eines Schreibbefehls auf das Register X von Planet A auf die Replika von Planet B. Die beiden Planeten haben einen Abstand von d . Da der $read$ -Befehl hin und zurück muss, dauert die Ausführung $2 \cdot \frac{d}{v_B}$.

- A) (5 Punkte) Angenommen das System besteht aus den 3 Planeten A, B und C , die sich relativ zueinander nicht bewegen. Die Distanzen zwischen den Planeten sind: $d_{AB} = v_B$, $d_{AC} = \sqrt{2} \cdot v_B$ und $d_{BC} = v_B$. Zeichne den zeitlichen Verlauf der Operationen in Tabelle 1 als timeline. Erfüllt diese Abfolge von Operationen das *read-your-writes*-Konzept?

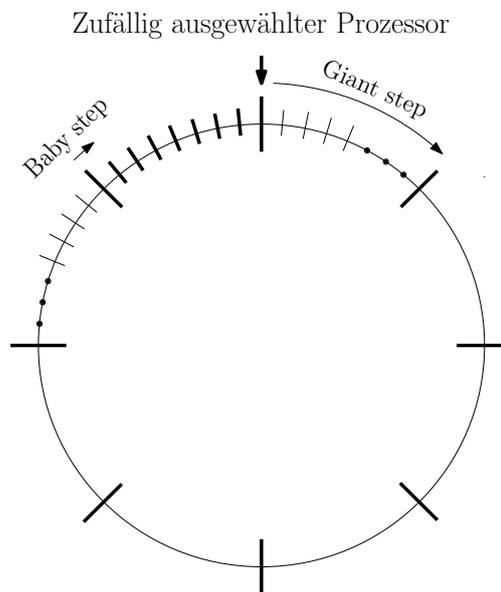
Planet	Zeitpunkt	Operation
A	0	$write(X, x_1)$
A	1	$read_B(X)$
A	3	$write(X, x_2)$
A	4	$read_C(X)$

Tabelle 1: Die in Aufgabe A) ausgeführten Operationen.

- B) (6 Punkte) Das System besteht nun aus N Planeten, die sich relativ zueinander nicht bewegen. Welche der folgenden Konsistenzmodelle werden von diesem System in jedem Fall erfüllt: *monotonic-read*, *monotonic-write* und *sequential consistency*.
- C) (3 Punkte) Zusätzlich betrachten wir nun ein Raumschiff, das einen Server mit Replikas mit sich führt. Das Raumschiff fliegt mit einer Geschwindigkeit von $v_R < v_B$. Welche der Konsistenzmodelle aus Aufgabe B) sind nun noch erfüllt.
- D) (3 Punkte) Welche Konsistenzmodelle werden noch erfüllt, falls das Raumschiff aus Aufgabe C) auch mit Geschwindigkeiten $v_R > v_B$ unterwegs sein kann.

13 "Chord"-inspiriertes Quorum System (19 Punkte)

Betrachten Sie das folgende Quorum System. Die n Prozessoren sind kreisförmig angeordnet. Ein Quorum wird folgendermassen ausgewählt: Starte bei einem zufälligen Prozessor und mache von diesem $\log_2 n - 1$ grosse Schritte (giant steps) mit der Schrittlänge $n/\log_2 n$. Danach mache $n/\log_2 n$ kleine Schritte (baby steps) mit Schrittlänge 1. Alle Prozessoren die auf diese Weise besucht wurden sind Teil des Quorums. In der Zeichnung ist ein solches Quorum System angedeutet: Die breiten Striche stehen für ausgewählte Prozessoren. Sie können davon ausgehen, dass n sowohl eine Zweierpotenz, als auch ein Vielfaches von $\log_2 n$ ist.



- A) (4 Punkte) Warum ist dieses Quorum System korrekt?
- B) (7 Punkte) Was sind die Load, Work und die Resilience dieses Systems?
- C) (4 Punkte) Was ist die asymptotische Fehlerwahrscheinlichkeit (failure probability) dieses Systems?
- D) (4 Punkte) Sie werden damit beauftragt, die Load dieses Systems zu verringern. Wie wählen sie die Schrittgrössen für eine optimale Load?