

## Egzamin z systemów rozproszonych

3 lutego 2011

Część: I (piszą wszyscy)

Czas pracy: 1.5 h

Punkty możliwe do zdobycia: 25

**Każde pytanie na osobnej kartce.**

**Pytanie 1 (5 pkt).** Dostęp do aktualnego czasu globalnego (np. UTC) na każdej maszynie jest przydatny w wielu systemach i algorytmach rozproszonych.

- Wyjaśnij dlaczego zegary lokalne różnych maszyn mogą pokazywać różny czas (1 pkt).
- Przyjmując, że maksymalna różnica czasu lokalnego na dwóch maszynach tolerowana przez pewien system rozproszony to  $\delta$ , podaj maksymalny tolerowany czas pomiędzy dwoma kolejnymi resynchronizacjami zegara lokalnego pojedynczej maszyny (do czasu globalnego). Wprowadź dodatkowe oznaczenia specyfikujące zachowanie zegara lokalnego maszyny (2.5 pkt).
- Podaj co najmniej dwa powody, dla których czas resynchronizacji w używany w praktyce powinien być krótszy niż obliczony wyżej czas maksymalny (1.5 pkt).

**Pytanie 2 (5 pkt).** Formalne modele spójności są rodzajem kontraktu pomiędzy twórcami a użytkownikami systemu rozproszonego. Kontrakt taki precyzuje, jakie wyniki mogą dać operacje użytkownika, gdy operacje te są wykonywane współbieżnie z operacjami innych użytkowników lub poprzez różne serwery systemu.

- Opisz na czym polega model spójności sekwencyjnej (ang. *sequential consistency*) (1 pkt).
- Podaj przykład operacji (np. używając notacji z wykładu), które są spójne sekwencyjnie, ale które nie są ukończone na wszystkich maszynach w kolejności, w której były rozpoczęte, tj. nie są ściśle spójne (ang. *strictly consistent*) (1 pkt).
- Opisz model spójności przyczynowo-skutkowej (ang. *causal consistency*) (1 pkt).
- Podaj przykład operacji, które są spójnie przyczynowo-skutkowo, ale nie sekwencyjnie (1 pkt).
- Dwa powyższe modele spójności są zorientowane na dane (ang. *data centric*). Krótko scharakteryzuj modele spójności zorientowane na klienta (ang. *client centric*) i wymień (bez opisywania) dwa przykłady takich modeli (1 pkt).

**Pytanie 3 (5 pkt).** Istnieje wiele protokołów zapewniających spójność zreplikowanych danych. Popularną rodziną takich protokołów są protokoły, w których jedna z replik pełni funkcję nadrzędną (ang. *primary-based protocols*).

- Podaj najsilniejszy model spójności (zorientowany na dane), jaki mogą zapewnić te protokoły (1 pkt).
- W kilku zdaniach lub punktach opisz przykładowy protokół (3 pkt).
- Wyjaśnij różnicę pomiędzy blokującym i nieblokującym wariantem protokołu. Wymień co najmniej jedną wadę i zaletę każdego z wariantów (1 pkt).

**Pytanie 4 (5 pkt).** Odporność na błędy jest bardzo istotną cechą systemów rozproszonych.

- Wymień trzy przykładowe rodzaje błędów ze względu na zachowanie komponentów systemu w trakcie ich wystąpienia (ang. *failure models*). Opisz w jednym-dwóch zdaniach, na czym one polegają (1.5 pkt).
- W kilku zdaniach opisz na czym polega dostępność systemu (ang. *availability*) (1.5 pkt).
- W kilku zdaniach opisz, czym różni się niezawodność systemu (ang. *reliability*) od jego dostępności (2 pkt).

**Pytanie 5 (5 pkt).** Rozproszone protokoły zatwierdzania (ang. *distributed commit protocols*) pełnią istotną rolę w wielu systemach rozproszonych, poczynając od rozproszonych systemów plików do rozproszonych baz danych.

- Podaj w jednym zdaniu, co jest zadaniem protokołów zatwierdzania (0.5 pkt).
- Rozróżnij stany i przejścia między nimi dla wszystkich rodzajów procesów protokołu zatwierdzania dwufazowego (ang. *two-phase commit*) (1 pkt).
- W punktach wyjaśnij, co dzieje się w każdym przejściu podczas normalnej operacji (2 pkt).
- Wyjaśnij, co się dzieje w stanie numer 2 podczas awarii każdego z procesów (1.5 pkt).

## Distributed Systems: Final Exam

February 3, 2011

Part: I (everybody should write)

Exam duration: 1.5 h

Total points: 25

**Each question should be answered on a separate sheet.**

**Question 1 (5 pts).** Many distributed systems and algorithms make use of the fact that each machine has access to global time (e.g., UTC).

- Explain why the local clocks of different machines can display different time (1 pt).
- Assume that the maximal difference between the local time of two machines that can be tolerated by some distributed system is equal to  $\delta$ . Compute the maximal possible interval between two consecutive resynchronizations of a single machine with the global time. If necessary, introduce any additional symbols describing the parameters of the machine's local clock (2.5 pts).
- Give at least two practical reasons for having the interval between consecutive resynchronizations shorter than the maximal value computed in the previous point (1.5 pt).

**Question 2 (5 pts).** Formal consistency models constitute a form of a contract between the developers and users of a distributed system. Such a contract specifies what results a user can expect from his operations if these operations are executed concurrently with operations of other users or through different servers running the system.

- Explain the *sequential consistency* model (1 pt).
- Using the notation introduced during the lectures, give an example of operations that are sequentially consistent, but at the same time, they are not completed in the actual order they were issued, that is, they are not *strictly consistent* (1 pt).
- Explain the *causal consistency* model (1 pt).
- Give an example of operations that are causally consistent, but not sequentially consistent (1 pt).
- The above consistency models are data centric. Briefly explain another family of consistency models: client-centric models. Give two examples of such models (1 pt).

**Question 3 (5 pts).** Many various protocols exist for maintaining the consistency of replicated data. A common family of such protocols are primary-based protocols. In primary-based protocols, one of the replica nodes acts as a special node, a so-called *primary*.

- Give the strongest data-centric consistency models primary-based protocols can provide (1 pt).
- In a few sentences or points, describe a sample primary-based protocol (3 pts).
- Explain the difference between the blocking and nonblocking variant of the protocol. Give at least one advantage and one disadvantage of each variant (1 pt).

**Question 4 (5 pts).** Fault tolerance is an important aspect of distributed systems.

- Give three examples of different failure models. In one-two sentences, describe each of these models (1.5 pt).
- In a few sentences, describe what high system availability denotes (1.5 pt).
- In a few sentences, describe how reliability differs from availability (2 pts).

**Question 5 (5 pts).** Distributed commit protocols are fundamental to many distributed systems, from distributed file systems to distributed data bases.

- In one sentence, summarize the objective of distributed commit protocols (0.5 pt).
- Draw the states and state transitions for all types of processes participating in a two-phase commit protocol (1 pt).
- In points, describe what happens during each state transition, assuming that there are no failures (2 pts).
- Explain what happens in state two upon a failure of each process type (1.5 pts).