



DS | Exam

(15 bodov) Vysvetlite dôvod zavedenia konzistencie zameranej na klienta v distribuovanom systéme a uveďte konkrétne konzistentné modely. Uvažujme nad službou poštovej schránky pre mobilného používateľa, ktorá je implementovaná ako súčasť rozsiahlej distribuovanej databázy. Ktorý konzistentný model zameraný na klienta by bol najvhodnejší? Odpoveď zdôvodnite.

(15 points) Explain the reason for establishment of client-centric consistency in a distributed system and list the individual consistency models. Consider a personal mailbox service for a mobile user, implemented as part of a wide-area distributed database. What kind of client-centric consistency model would be most appropriate? Explain why.

- Pre službu poštovej schránky mobilného používateľa je najvhodnejší model **Monotonic Reads**. Ten zabezpečuje, že keď používateľ raz zobrazí e-mail (napr. z jedného zariadenia), všetky následné prístupy (napr. z iného zariadenia) odrážajú rovnaký alebo novší stav schránky. Zabráňuje čítaniu neaktuálnych údajov zo zastaraných replík a poskytuje konzistenciu pre používateľa.

(15 bodov) Vysvetlite čo je to konsenzus v skupine replikovaných procesov v distribuovanom systéme odolnom voči výpadkom (crash failure) a aký je dôvod pre jeho zavedenie. Vysvetlite konzistentný protokol Raft.

(15 points) Explain what consensus is in a group of replicated processes in a distributed system tolerant to crash failures, and the reason for its establishment. Explain consistency protocol Raft.

- Konsenzus v distribuovanom systéme s replikovanými procesmi zabezpečuje, že všetky procesy, ktoré nie sú chybné, sa dohodnú na postupnosti vykonávaných operácií.

- Protokol Raft organizuje tieto repliky do modelu 1 leader servera. Raft poskytuje jednoduchosť a transparentnosť, vďaka čomu sa ľahšie implementuje ako Paxos.

(10 bodov) Aká je úloha midlvéru v distribuovanom systéme?

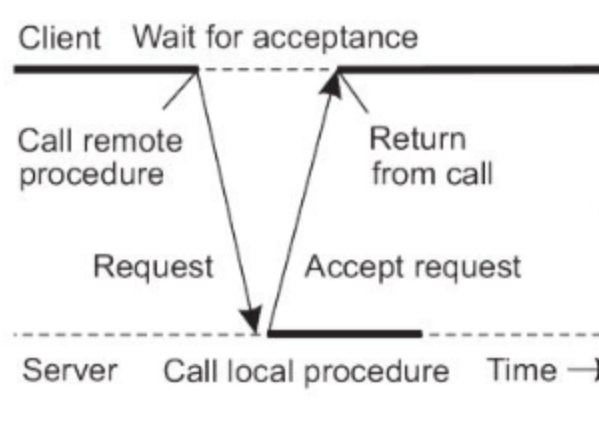
(10 points) What is the role of middleware in a distributed system?

- Middleware v distribuovaných systémoch je softvérová vrstva, ktorá zabezpečuje komunikáciu medzi komponentami, uľahčuje zdieľanie zdrojov, bezpečnosť a obnovu distribučného systému.
- REST API, RabbitMQ, Kafka, JDBC, ...

(10 bodov) Predpokladajme, že v sieťovom podsystéme môžete používať iba blokujúce operácie priebežnej synchronnej komunikácie. Ako by ste pomocou týchto synchronných operácií implementovali nové neblokujúce operácie priebežnej asynchronnej komunikácie?

(10 points) Suppose that you could make use of only transient synchronous blocking communication primitives in the network subsystem. How would you implement new primitives for transient asynchronous non-blocking communication using those synchronous ones?

- Bolo by nutné použiť multithreading - vytvoriť asynchronnu komunikáciu tým, že sa budú na viacerých vláknach vykonávať súbežné procesy (odosielanie/prijímanie) a nebudú sa teda navzájom blokovať
 1. Použije sa samostatné vlákno alebo proces.
 2. Výsledok sa uloží do frontu.
 3. Upozornenie hlavnej aplikácie na výsledok pomocou callbacku.



(10 bodov) Je striktné nutné potvrdiť prijatie každej správy, aby sme dosiahli plne usporiadaný multicast pomocou Lamportových časových značiek?

Odpoveď zdôvodnite.

(10 points) To achieve totally-ordered multicasting with Lamport timestamps, is it strictly necessary that each message is acknowledged? Explain why.

- Potvrdenie prijatia každej správy je potrebné, pretože zabezpečuje, že všetky procesy v distribuovanom systéme sa nakoniec dohodnú na poradí správ synchronizáciou svojich logických hodín a overením prijatia a spracovania správy.

(10 bodov) Ako spraviť aby server ktorý má viac vlákien akceptoval správy z rôznych protokolov?

(10 points) How to make multithread server accept messages from variety of protocols?

- Použije sa middleware (napr. message broker) na štandardizáciu správ
- Thread Pooling - určí sa dispatcher thread, ktorý prideliuje operácie zvyšným worker threadom