Numerical Solutions for Distributed Systems

Question 1

Suppose that the time to do a null RPC (i.e., 0 data bytes) is 2.0 msec, with an additional 2.5 msec for every 1K of data. How long does it take to read 42K from the file server in a single 42K RPC? How about as 42 1K RPCs?

Solution

Given Information:
- Null RPC Time: 2.0 milliseconds (msec)
- Additional Time per 1K of data: 2.5 msec
- Data Size: 42K

We'll solve this in two parts.

Part A: Single 42K RPC

1. Calculate the base time for a null RPC: 2.0 msec.
2. Calculate the additional time for 42K of data:
   - Since it takes 2.5 msec for every 1K of data, for 42K we get:
     42 × 2.5 msec = 105 msec
3. Total time for a single 42K RPC:
   - Add the null RPC time and additional data time:
     2.0 msec + 105 msec = 107 msec

Answer for Part A: 107 msec

Part B: 42 RPCs of 1K Each

1. Time for each 1K RPC:
   - Null RPC time = 2.0 msec
   - Additional time for 1K = 2.5 msec
   - Total time per 1K RPC = 2.0 + 2.5 = 4.5 msec
2. Total time for 42 RPCs of 1K each:
   - Since we have 42 of these RPCs:
     42 × 4.5 msec = 189 msec

Answer for Part B: 189 msec

Question 2

Consider the behaviour of two machines in a distributed system. Both have clocks that are supposed to tick 12000 times per millisecond. One of them actually does, but the other ticks

only 1000 times per millisecond. If UTC updates come in once a minute, what is the maximum clock skew that will occur?

Solution

Given Information:
- Clock ticks per millisecond:
  - Machine A: 12000 ticks (accurate clock)
  - Machine B: 1000 ticks (slower clock)
- Time interval between UTC updates: 1 minute (or 60,000 milliseconds)

Find the Maximum Clock Skew:

1. Clock Ticks in 1 Minute (60000 ms)
   - Machine A (accurate clock): 12000 × 60000 = 720,000,000 ticks
   - Machine B (slower clock): 1000 × 60000 = 60,000,000 ticks

2. Milliseconds Passed According to Each Machine
   - For Machine A (accurate clock): 60000 ms (since it's accurate).
   - For Machine B (slower clock):
     - 60,000,000 ticks ÷ 12000 ticks/ms = 5000 ms (only 5 seconds).

3. Clock Skew (Difference) After 1 Minute
   - Machine A shows 60000 ms, while Machine B shows only 5000 ms.
   - So, the maximum clock skew is:
     60000 - 5000 = 55000 ms = 55 seconds

Answer: Maximum clock skew is 55 seconds.

Question 3

A process with transaction timestamp 100 needs a resource held by a process with transaction timestamp 150. Compare the results when:
i. Wait-die deadlock prevention algorithm is used.
ii. Wound-wait deadlock prevention algorithm is used.

Solution

Part A: Wait-Die Algorithm

In the Wait-Die algorithm:
- Older process (with the smaller timestamp) can wait for the younger process.
- Younger process (with the larger timestamp) will be killed (or aborted) if it tries to wait for an older process.

Solution:
- Process with timestamp 100 is older than the process with timestamp 150.

- Since the older process (100) wants to access a resource held by the younger process (150), Process 100 is allowed to wait.

Answer: Process 100 waits for Process 150 in the Wait-Die algorithm.

Part B: Wound-Wait Algorithm

In the Wound-Wait algorithm:
- Older process (with the smaller timestamp) will "wound" (force the abort of) the younger process if it needs a resource held by the younger process.
- Younger process (with the larger timestamp) can wait if it needs a resource held by the older process.

Solution:
- Process with timestamp 100 (older) needs a resource held by Process 150 (younger).
- Since the older process requests a resource from the younger, Process 100 wounds (forces an abort of) Process 150.

Answer: Process 150 is aborted (wounded) by Process 100 in the Wound-Wait algorithm.