

Versão A - read committed

Nos nossos experimentos, essa foi a melhor versão: rodou em menos tempo e os clientes tiveram menos tentativas de reservas até conseguir sua passagem. O fato de menor quantidade de tentativas deu-se pelos passos serem executados como uma transação só que, após bloquear a passagem, impede que outras transações acessem aquela passagem. A reserva é tratada como uma única transação, garantindo consistência nos dados durante todo o processo e os dados podem ser acessados para leitura por outras transações. No entanto, pode haver bloqueio de recursos durante a execução da transação, o que pode causar atrasos para outras transações que necessitem dos mesmos recursos. Se ocorrerem erros ou falhas durante qualquer um dos três passos da reserva, a transação precisa ser revertida, o que pode causar perda de tempo e recursos.

Versão B - read committed

O Passo 2 pode ser executado de forma independente da transação, o que pode melhorar a performance e evitar bloqueios desnecessários. Assim como na versão A com read committed, os dados acessados no passo 1 podem ser lidos por outras transações e não há o risco de leituras sujas.

Versão A - serializable

A reserva é tratada como uma única transação, dessa forma a consistência nos dados é garantida, mas há bloqueio de recursos durante a execução da transação, causando atrasos para outras transações. Mantém o mais alto nível de isolamento, impedindo a ocorrência de leituras sujas, leituras não repetíveis e escritas fantasmas, porém o nível serializável gera problemas de concorrência e baixa escalabilidade em sistemas com alta concorrência de transações. Garante uma visão consistente dos dados durante toda a transação.

Versão B - serializable

O Passo 2 pode ser executado de forma independente e fora do contexto transacional, melhorando a performance e evitando bloqueios desnecessários, no entanto ser dividida em duas transações pode levar a inconsistências caso ocorram falhas entre as duas transações. Mantém o mais alto nível de isolamento, impedindo a ocorrência de leituras sujas, leituras não repetíveis e escritas fantasmas, porém pode levar a problemas de concorrência e baixa escalabilidade em sistemas com alta concorrência de transações.