|  |
| --- |
| Wojciech Ganobis Lista 8  Redaktor: |

1. Jeśli serwer nie będzie bezpieczny, nie będziemy mogli mu ufać że działa dobrze i że ktoś złośliwy go nie zmodyfikował.
2. Serwer może zobaczyć taką awarie jako zaprzestanie produkcji danych wyjściowych, może on pomyśleć, że serwer działa wolno lub komunikacja tymczasowo sie przerwała.
4. Nie.
5. W każdym rzędzie kręgów może zawieźć jeden element. Przykład najgorszego przypadku to jeśli wszystkie sześć elementów znajduje się u góry, wszystkie kolumny zawodzą, dwa z trzech końcowych wyników będą poprawne, więc możemy przetrwać sześć awarii.
6. Tak. Mogłaby zamaskować do dwóch usterek na grupę urządzeń.
7. a i b są co najmniej jednokrotne. NIc się nie stanie jeśli będziemy wielokrotnie wykonywać tą akcje. Natomiast w przypadku c użytkownik będzie musiał podjąć działania, aby naprawić problem
9. Wysyłanie scen filmowych, gdzie na początku każdej sceny jt numer sceny.
10. NIe. Nie ma potrzeby aby warstwa komunikacyjna utrzymywała kopie komunikatu, ponieważ konieczna jest dostępnośc danych na poziomie aplikacji.
11. To zależy od liczby procesów. Jeżeli procesy są klonowane aby zwiększyć odpornośc na uszkodaenia, wystarczy kilka replik.Wtedy skalowalnośc nie stanowi problemu. Jednak gdy tworzymy grupy różnych procesów może pojawić się problem.
12. Nie możemy udzielać takich gwarancji, podobnie jak w przypadku gwarancji ,że serwer faktycznie wykonał operację po wysłaniu potwierdzenia do klient. Jednak stopień tolerancji na uszkodzenia poprawia się za pomocą zastosowania schematów rozsyłania niepodzielnego.
13. Synchronizacja wynika z pojedynczego rozsyłania niezależnie od tego czy, są one zamówione przez FIFO czy są przyczynowe. Zmiany widoku mają miejsce jako specjalne wiadomości rozsyłania, które również muszą być odpowiednio uporządkowane.
14. Uporządkowanie1(m1, m2, m3, m4)  
    Uporządkowanie2(m1, m3, n2, m4)  
    Uporządkowanie3(m1, m3, m4, m2)  
    Uporządkowanie4(m3, m1, m2, m4)  
    Uporządkowanie5(m3, m1, m4, m2)  
    Uporządkowanie6(m3, m4, m1, m2)

17. Nie. Należy zauważyć, że proces odzyskiwanie który nie brał udziału w ostatecznej decyzji, powrócił do stanu który jest zgodny z ostatecznym wyborem dokonanym przez pozostałe procesy.
18. Należy też rejestrować inne zdarzenia.
19. Rejestrowanie zawiera każdą operacje odczytu i zapisu która miała miejsce w ramach transakcji. Gdy awaria nastąpi to rejestrowanie może otworzyć zapis aż do ostatniej zarejestrowanej operacji. Ponowne odtworzenie rejestrowania jest w rzeczywistości przeciwieństwem cofania, które ma miejsce, gdy transakcja musi zostać przerwana.
20. Nie zawsze. Punkty kontrolne są stosowane tylko w przypadku odzyskiwania lokalnego, a na przykład serwer baz danych nie ma takiej potrzeby.
21. Głównym powodem jest to, że odzyskiwanie danych jest całkowicie lokalne.