|  |
| --- |
| Wojciech Ganobis Lista 3  Redaktor: |

1. W jednowątkowym serwerze czas wykonania jednego zapytania to 15ms lub 90ms(w zależności od tego czy dane są w pamięci podręcznej). Operacji dyskowych trzeba wykonać zazwyczaj około ⅓ więc średni czas to: ⅔\*15ms + ⅓\*90ms = 40ms. Natomiast w wielowątkowym serwerze ten czas to 15ms. W sekundę możemy więc wykonać 25 zamówień w systemie jednowątkowym lub 66 zamówień w serwerze wielowątkowym.
2. Tak, wątki potrzebują mieć własny stos. Posiadanie więc zbyt wielu wątków może powodować bardzo duże zapotrzebowanie na pamięć i serwer nie będzie działał poprawnie.
3. Tak, nie zawsze potrzebujemy wprowadzać dodatkową złożoność.
4. Takie przypisanie ogranicza się do posiadania jedynie wątków na poziomie jądra, co prowadzi do tego, że dużą część wzrostu wydajności z posiadania wątków na pierwszym miejscu jest tracona. Proces lekki to pośredni poziom wątków. Każde zadanie ma przynajmniej jeden proces lekki do którego podłączone są wątki poziomu użytkownika.
5. W takim schemacie ograniczamy się do wątków na poziomie użytkownika, co oznacza, że każde wywołanie blokujące systemowe, blokuje cały proces.
6. Zaczynamy z jednym procesem lekkim który wybiera wykonywalny wątek. Po znalezieniu, tworzy nowy lekki proces który szuka kolejnego wykonywalnego wątku który mógłby wykonać. Jeżeli nie znajdzie takowego proces umiera.
7. Tak. Serwer kontroluje sprzęt, a aplikacja wysyła żądania aby sterować sprzętem.
8. Można zastosować techniki kompresji, wtedy przepustowość się zmniejszy, przez co poradzimy sobie ze skalowalnością numeryczną.
9. Tak. Załóżmy, że mamy dwa obiekty: replikowany A oraz niereplikowany B. Jeśli A składa się z n replik, B zostanie wywołane przez każdą replikę A, a powinno zostać wywołane raz.
10. Jedną z zalet jest, że oddzielne procesy chronią siebie nawzajem co może być konieczne, jak w przypadku superserwera obsługującego niezależne usługi.Z drugiej strony tworzenie procesu jest drogą operacją którą można zaoszczędzić używając serwera wielowątkowego. Dodatkowo jeżeli procesy muszą się komunikować użycie wątków jest dużo tańsze.
11. Gniazda to zestaw narzędzi programistycznych do komunikacji międzyprocesowej.
12. Użycie mikrojądra rozwiąże ten problem.
13. Jeśli serwer nie przechowuje, żadnych informacji o kliencie to serwer jest bezstanowy, ponieważ to nie serwer a warstwa transportowa przechowuje informacje o stanie klienta.
14. Jest to serwer bezstanowy. Serwer ten nie przechowuje informacji o kliencie jak było już powiedziane w zadaniu 13.
15. Rozwidlanie w systemach UNIX oznacza, że pełny obraz rodzica jest kopiowany do potomka. Pierwszym krokiem jest zarezerwowanie miejsca dla nowego procesu, stworzenie go i utworzenie odpowiedniej mapy procesów oraz pamięci. Po wykonaniu tych operacji można rodzica skopiować, a potomka aktywować.
17. Można stworzyć proces Q, który obsługuje zdalnie zapytania dla F. Procesowi P oferujemy taki sam interfejs co dla F. Finalnie Q działa już jako serwer plików.