lista 3 - "Systemy Rozproszone 2020"

Łukasz Klasiński

1. W tym zadaniu należy porównać czytanie pliku za pomocą jednowątkowego serwera plików i serwera wielowątkowego. Otrzymane zamówienia na pracę, skierowanie go do wykonania i uporanie się z resztą niezbędnego przetwarzania zajmuje 15 ms, pod warunkiem że potrzebne dane są w pamięci podręcznej utrzymywanej w pamięci operacyjnej. Jeżeli powstaje konieczność wykonania operacji dyskowej, co stanowi jedną trzecią ogółu zamówień, trzeba dodatkowych 75 ms, podczas których wątek jest uśpiony. Ile zamówień na sekundę może obsłużyć serwer jednowątkowy? A serwer wielowątkowy?

1. Serwer jednowątkowy: Jako że co trzecie zamówienie wymaga operacji dyskowej to widać, że do przetworzenie 3 kolejnych zamówień zajmuje

$$15 + 15 + 15 + 75 = 120ms120ms * 8 = 960ms$$

   Zatem w 1 sekundzie jesteśmy w stanie wykonać $8 * 3 + 2 = 26$ zamówień.
2. Serwer wielowątkowy: Załóżmy że nasz serwer ma n wątków. Dodatkowo dla uproszczenia powiemy, że $\frac{1}{3}n$ wątków jest dedykowania dla zamówień wymagających operacji dyskowych. W takim razie w trakcie $1000ms$, $\frac{2}{3}n$ dedykowanych do obsługi zwykłych zamówień procesów będzie w stanie przetworzyć $\frac{2}{3}n * 66 = 44n$ zamówień. Dodatkowo reszta procesów dedykowanych dla operacji dyskowych, wykonają po 11 zamówień, czyli łącznie

$$44n + \frac{1}{3}n * 11 = n(44 + \frac{11}{3})$$

   Oczywiście jest to prawdziwe tylko kiedy mamy dostępne co najmniej 3 wątki.

2. Czy dałoby się uzasadnić ograniczenie liczby wątków w procesie serwera?

Tak, jestem w stanie przytoczyć dwa powody: 1. Jeśli liczba wątków przekracza możliwości obliczeniowe procesora (liczbę jego wątków), to zamiast przyspieszenia działania programu widoczne jest znaczące spowolnienie, bo większość czasu jest spędzane na przełączaniu kontekstu zamiast samego ich liczenia. Oczywiście przy założeniu że nasze wątki są cały czas aktywne.

2. Za duża liczba wątków, powoduje ze dany proces ma przydzielony znacznie większy czas procesora tylko dla siebie, przez co gdy inne procesy nie mają tak dużej liczby, to nie będą one otrzymywać wystarczającej ilości zasobów aby poprawnie (szybko) działać. Jeśli wszystkie procesy mają dużą liczbę wątków, to wracamy do punktu 1.

3. W treści rozdziału zawarto opis wielowątkowego serwera plików, podkreślając jego zalety w porównaniu z serwerem jednowątkowym i serwerem działającym niczym maszyna o skończonej liczbie stanów. Czy istnieją okoliczności, w których serwer jednowątkowy może okazać się lepszy? Jeśli tak, to podaj przykład. Jeżeli nie, to uzasadnij dlaczego.

Tak, może okazać się lepszy. Przykład: strony www. Okazuje się że serwery obsługujące wiele zapytań http gdyby miały procesory wielowątkowe, to musiałyby dla każdego takiego zapytania tworzyć nowy wątek. Przez to tworzy się niekontrolowana ilość wątków, które jak wiadomo z 2 zadania przy przekroczeniu ilości rdzeni mogą znacząco spowolnić serwer oraz wymagają dużej ilości pamięci operacyjnej. Dlatego zwykle używa się dużej ilości serwerów jednowątkowych, które dzielą się zamówieniami a nadmiarowe mogą kolejkować zamiast `wciskać` w nowo stworzone procesy.

4. Statyczne przypisanie tylko jednego wątku do procesu lekkiego nie jest zbyt dobrym pomysłem. Dlaczego?

Procesy lekkie są pośrednikami miedzy wątkami użytkownika oraz wątkami jądra. Każde zadanie ma przynajmniej jeden process lekki do którego podłączone są wątki użytkownika. W szczególności do jednego procesu lekkiego może być podłączony więcej niż jeden wątek użytkownika. W takim razie jeśli ustalimy, że process lekki ma tylko jeden wątek, to wtedy wątki użytkownika do niego podłączone będą mogły działać na fizycznie tylko jednym wątku, więc utworzy się wąskie gardło na danym lekkim procesie.

5. Posiadanie tylko jednego procesu lekkiego w procesie też nie jest zbyt dobrym pomysłem. Dlaczego?

Jak wiadomo wątki użytkownika mogą mieć podłączone więcej niż jeden proces lekki, zatem jeśli nie pozwolimy na więcej niż jeden proces lekki to możemy zagłodzić wątki danego procesu.

7. W systemie X określa się terminal użytkownika jako serwer goszczący, natomiast o aplikacji mówi się że jest klientem. Czy ma to sens?

Jeśli przykładowo chcemy pracować zdalnie, to na naszym lokalnym komputerze działałby jako serwerem X, natomiast program który wykonuje się zdalnie na innym komputerze (serwerze) byłby nazwany klientem, co troche zaprzecza temu że pracujemy na serwerze.

10. Budowanie serwera współbieżnego na zasadzie namnażania procesów ma pewne zalety i wady w porównaniu z serwerami wielowątkowymi. Wymień kilka.

Wady: * Duży koszt utworzenia nowego procesu

- Ciężko kontrolować liczbę procesów
- Duży koszt usuwania procesów (wydajność spada kiedy mamy dużo `małych` zapytań)
- Jeśli kilka połączeń korzysta ze wspólnych zasobów, to każdy proces ma swoją kopie (zamiast je dzielić)

Zalety: * Prosty program obsługi serwera

- Zrównoleglamy połączenia zamiast zapytań, więc jeśli są one dłuższe to mamy lepszą wydajność

13. Czy serwer utrzymujący z klientem połączenie TCP/IP jest pełnostanowy czy bezstanowy?

Jest pełnostanowy, ponieważ protokół TCP wymaga ustalenia sesji połączenia z klientem która zawiera informacje na temat jego ip oraz portów, zatem nie możemy mówić tu o bezstanowym serwerze.

14. Wyobraź sobie serwer Sieci, który działa na tablicy z odwzorowanymi adresami IP ostatnio odwiedzanych przez klienta stron WWW. Gdy klient łączy się z serwerem, ten poszukuje klienta w swojej tablicy i jeśli go znajdzie, to zwraca mu zarejestrowaną stronę. Czy serwer taki jest pełnostanowy czy bezstanowy?

Serwer jest pełnostanowy, ponieważ jeśli stan serwera się zmieni (np. wyczyści się historia przeglądani), to klient przestanie mieć dostęp do jego ostatnio odwiedzanych stron (jego stan też się zmieni).