

lista 2 - “Systemy Rozproszone 2020”

Łukasz Klasieński

1. Jeśli klient i serwer znajdują się w odległych od siebie miejscach, zauważamy, że opóźnienie ma zasadniczy wpływ na wydajność. Jak można temu zaradzić?

Możemy zbudować pośrednika między serwerem i klientem, który będzie znajdował się w bliższej odległości, dzięki czemu potrzebne dane mogą być w nim buforowane i wysyłane do klienta zamiast ciągłego wysyłania zapytań do serwera.

2. Co to takiego trzywarstwowa architektura klient-serwer?

Przyjmuje się następujące warstwy:

1. poziom interfejsu użytkownika (KCK) - Jest to bezpośredni interfejs aplikacji, która obsługuje połączenie z serwerem i wykonywanie na nim operacji
2. poziom przetwarzania - aplikacja(je), która przetwarza zapytania i pobiera odpowiednie dane z trzeciego poziomu - np. przeglądarka jeśli klient chce wejść na daną stronę html musi odpowiednio przygotować odpowiednie zapytanie w odpowiednim protokole, następnie wysłać to zapytanie i odpowiednio przetworzyć otrzymane informacje, które następnie są przesyłane do warstwy KCK.
3. poziom danych (bazy danych) - jest to jakiś silnik który zarządza danymi na serwerze, oferuje jakiś interfejs, który umożliwia dostęp do danych które potrzebuje dana aplikacja z drugiej warstwy.

3. W czym zawiera się różnica między rozproszeniem pionowym a poziomym?

1. Rozproszenie pionowe - poszczególne warstwy oprogramowania architektury wielowarstwowej (np. trzywarstwowa z poprzedniego zadania) instalowane i uruchamiane są na różnych komputerach.
2. Rozproszenie poziome - rozproszeniu podlegają części składowe poszczególnych warstw (np w przypadku poziomu danych może oznaczać rozmieszczenie części danych na różnych serwerach). W ten sposób otrzymujemy współbieżne działanie na wielu serwerach na raz, równoważąc obciążenie serwerów.

4. Rozważmy łańcuch procesów P_1, P_2, \dots, P_n realizujących wielopiętrową architekturę klient-serwer. Proces P_i jest klientem procesu P_{i+1} , jak również zwraca odpowiedź procesowi P_{i-1} , lecz dopiero po otrzymaniu odpowiedzi od P_{i+1} . Na czym polegają główne trudności w tej organizacji, biorąc pod uwagę wydajność cyklu zamówienie-odpowiedź w procesie P_1 ?

- przede wszystkim taka organizacja jest bardzo podatna na usterki - wystarczy że jeden z procesów przestanie odpowiadać, a wszystkie pozostałe procesy które na nim polegają będą bezskutecznie oczekiwać na odpowiedź i wysyłać kolejne zapytania które nie zostaną zrealizowane.
- Ta organizacja jest podatna na duże opóźnienia, szczególnie kiedy mamy dużo procesów oraz przetworzenie zadania zajmuje każdemu procesowi sporo czasu. Widzimy że wtedy zapytania mogą się zacząć kolejkować i potrzebna jest odpowiednia logika tego które z nich mają większy priorytet i odpowiednie buffory na ich kolejkowanie.
- Ciężko taki system rozbudowywać - jedyną możliwością jest skopiowanie wszystkich procesów poza P_1 oraz P_n i podpięciem ich do kolejnego rzędu procesów. Oczywiście w takim rozwiązaniu wąskim gardłem jest P_n - jeśli nie jest w stanie obsługiwać wiadomości od jednocześnie k klientów, to taka rozbudowa niewiele da.

5. W strukturalnej sieci nakładkowej trasy komunikatów są wytyczne zgodnie z topologią nakładki. Co w tym podejściu stanowi istotną niedogodność?

Niedogodnością jest to, że trasy komunikatów w kolejności która wyznacza nam topologia niekoniecznie są optymalne - możemy mieć taką sytuację, że jakiś proces p_i chce wysłać wiadomość do procesu p_j , który jest w najdalszym punkcie na pierścieniu. Oczywiście musi taka wiadomość przejść przez wszystkie procesy na drodze $p_j \rightarrow p_i$. Ale może się okazać, że fizycznie proces p_j jest bardzo blisko do p_i , natomiast wiadomość przechodząc między poszczególne procesy będzie fizycznie krążyć po całym świecie zanim faktycznie dojdzie do p_j . Fajnie byłoby tak budować pierścień, aby procesy miały sąsiadów, którzy fizycznie są podpięci do sieci w jak najmniejszej od siebie odległości, tymczasem jest to robione losowo.

6. Rozważyć sieć CAN. Jak można by wytyczyć trasę komunikatu węzła o współrzędnych $[0.2, 0.3]$ do węzła o współrzędnych $[0.9, 0.6]$?

Wpierw węzeł $[0.2, 0.3]$ zapyta swoich sąsiadów, czy znają ścieżkę do $[0.9, 0.6]$. Otrzyma on odpowiedź od $[0.6, 0.7]$ oraz $[0.7, 0.2]$. Teraz zależnie który węzeł szybciej wysłał pozytywną odpowiedź, przysłała do niego komunikat przeznaczony do $[0.2, 0.3]$. W taki sposób wiadomość dotrze.

7. Biorąc pod uwagę, że węzeł w CAN zna współrzędne swoich bezpośrednich sąsiadów, rozsądną metodą trasowania mogłoby być przekazanie komunikatu do najbliższego węzła w kierunku docelowym. Skomentuj użyteczność tej metody.

Metoda może być skuteczna, ale tylko jeśli jesteśmy w stanie wyeliminować cykle które mogą się pojawiać. Ponadto najbliższy węzeł nie zawsze oznacza najkrótszą ścieżkę - możemy mieć wiele małych bloków, oraz jeden duży który w arytmetyki nie okaże się **najbliższy** i zamiast przekazać mu i docelowemu węzłowi, będziemy przekazywać do kilkunastu małych bloków, co znacząco spowolni czas dotarcia wiadomości do celu.

10. Nie każdy węzeł w sieci partnerskiej powinien być superpartnerem. Jakie rozsądne wymagania powinien spełniać superpartner.

Jako że superpartner jest pośrednikiem reszty partnerów w sieci partnerskiej, to rozsądnym powinien być taki wybór, aby sumaryczna długość ścieżki od każdego partnera do superpartnera była jak najmniejsza. Ponadto powinien on mieć lepszą przepustowość łączy oraz moc obliczeniową od zwykłego partnera, aby być w stanie przetwarzać wszystkie zapytania bez kolejkowania ich (aby nie był wąskim gardłem).

12. Podaj nieodparty (techniczny) argument przemawiający za tym, że polityce coś za coś (tit-for-tat) stosowanej w Bit torrents, wiele brakuje do optymalnej, jeśli chodzi o dzielenie się plikami w internecie.

Polityka ta zakłada, że dzielimy się tylko z tymi, którzy mają nam coś do zaoferowania, więc jeśli mamy w sieci kogoś, kto teoretycznie ma wszystkie pliki, to nie będzie miał nikogo kto mógłby mu coś zaoferować, więc nie będzie chciał się dzielić tym co ma, chociaż optymalniej byłoby dzielić się po to aby szybciej rozesłać dane wśród pijawek.