

Dokumentacja TIN

Temat

Sieciowy rozproszony system plików (peer-to-peer) pozwalający klientom na udostępnianie sobie plików.

Wymagania funkcjonalne

1	Pobranie pliku
1.1	Wysłanie zapytania do serwera o listę dostępnych do pobrania plików
1.2	Pobranie wybranego pliku
1.3	Przerwanie pobierania
2	Udostępnienie pliku
3	Zaprzestanie udostępniania pliku
4	Jednoczesne pobieranie i udostępnianie plików

Ogólny schemat działania

Ogólne założenia

Jeden serwer będzie obsługiwać dowolną liczbę klientów. Każdy z nich może udostępniać nowe pliki i pobierać wszystkie, które były dodane. Pobieralne dane są podzielone logicznie na bloki o stałej wielkości. Użytkownicy systemu muszą znajdować się w tej samej sieci, aby pominąć problem typowy dla aplikacji p2p, tzn. przebijanie NATów.

Cała funkcjonalność projektu polega na tym, że klienci pobierają pliki od siebie nawzajem, a serwer trzyma jedynie informacje o lokalizacji danych bloków wśród dostępnych użytkowników oraz pewne szczególne metadane. Pliki pobierane są

blokami, a nie w całości jak to ma miejsce przy ściąganiu plików poprzez przeglądarkę internetową. Kiedy użytkownik pobierze w całości dany blok, inni mogą ten blok od niego pobrać. Serwer znając rozmieszczenie poszczególnych bloków wśród hostów, wskazuje skąd należy pobierać kolejny żądany blok pliku.

Udostępnianie plików

Serwer przechowuje informacje na temat bloków (składowych plików) oraz klientów, którzy je udostępniają. Ma on też informacje na temat tego jak duża jest liczba klientów pobierających oraz klientów czekających na pobranie od wybranego hosta, co pozwoli na optymalizację czasu pobierania pliku. Serwer nie pozwala na udostępnianie dwóch różnych plików o tej samej nazwie.

Klient wysyła do serwera informacje o chęci udostępnienia pliku, wraz z informacjami takimi jak swój adres, nazwa tego pliku oraz jego rozmiar. Dodatkowo przesyła listę haszy dla każdego bloku, aby umożliwić późniejszą weryfikację. Serwer zapisuje u siebie te metadane oraz informacje o udostępnionych przez klienta blokach. W przypadku udostępnienia nowego pliku klient ten staje się pierwszym seederem, a od tego momentu plik jest gotowy do pobrania.

Pobieranie plików

Klient może wysłać zapytanie do serwera z prośbą o udostępnienie informacji o dostępnych plikach, a serwer w odpowiedzi wyśle mu ich listę wraz z rozmiarem, liczbą seederów i leecherów.

Klient wysyła do serwera informacje o nazwie pliku do pobrania oraz o numerach bloków, które pobrał do tej pory i które pobiera obecnie (jeśli takowe istnieją). Serwer na podstawie posiadanych informacji o hostach chcących udostępnić poszukiwane bloki danego pliku oraz o ilościach klientów pobierających lub czekających na pobranie od danego hosta wybiera od kogo dany klient ma pobrać blok pożądanego pliku. Bloki nie muszą być pobierane we właściwej kolejności.

Przy każdej prośbie o pobranie bloku, serwer w odpowiedzi przesyła odpowiednie dane hosta wraz z haszem, aby umożliwić klientowi sprawdzenie poprawności otrzymanego bloku.

Połączenie peer-to-peer

Klient po otrzymaniu od serwera adresu oraz portu peera, od którego powinien pobrać blok, wysyła do niego prośbę o udostępnienie bloku o danym indeksie z pliku o danej nazwie. Peer udostępniający blok sprawdza, czy dane się zgadzają. Jeśli nie, musi nastąpić odpowiednia obsługa w zależności od rodzaju błędu. Jeśli wszystko się zgadza, peer przesyła blok. Klient który go pobrał, sprawdza poprawność bloku haszując go i porównując otrzymany wynik z odpowiednim haszem który dostał od serwera.

Założmy, że do seedera, od którego klient chce pobrać blok, jest już kolejka. Wtedy klient musi czekać na przesłanie komunikatu od seedera, który powie, że teraz jest

jego kolej. Następnie klient wysyła potwierdzenie że jest gotowy do pobrania, po czym serwer przesyła mu blok.

W przypadku gdy klient czeka w kolejce, ale stwierdza, że już nie chce pobrać tego bloku, wtedy wysyła odpowiedni komunikat do peera i tym samym zwalnia miejsce w kolejce.

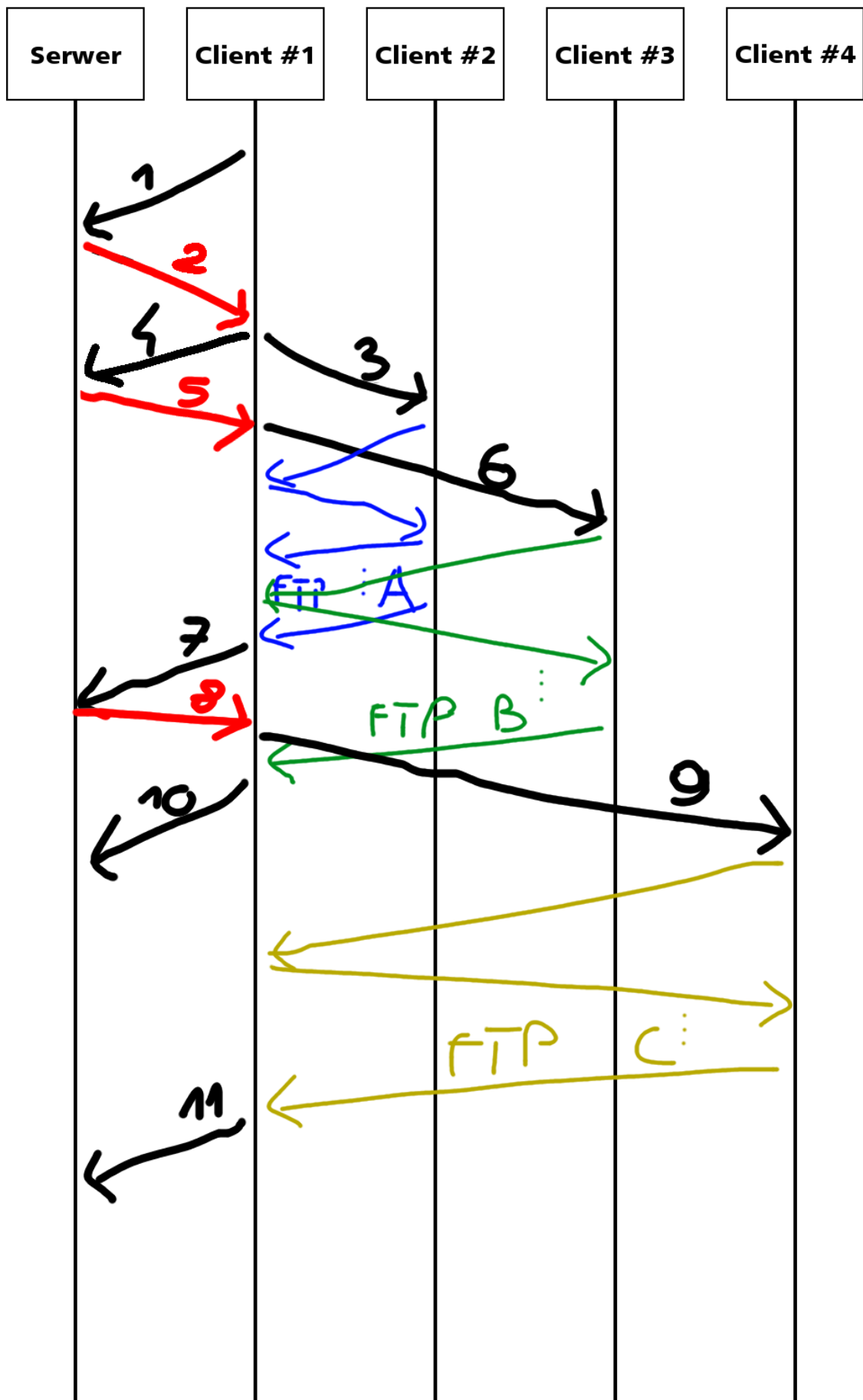
W trakcie pobierania/udostępniania

Klient wysyła regularne zapytanie rodzaju keep-alive do serwera informując go o swoim stanie (w trakcie pobierania, udostępniania), aby serwer mógł uaktualnić swoje listy peerów. W przypadku przekroczenia limitu czasowego, peer jest uznany za nieaktywny i połączenie zostaje zakończone.

Przykład

W poniższym przykładzie klient #1 chce ściągnąć plik składający się z 3 bloków (A, B, C). Klient na potrzeby przykładu ma 4 sockety do komunikacji z peerami/seederami - dwa na połączenia przychodzące i 2 na wychodzące, oraz kolejne dwa do komunikacji z serwerem.

1. Klient zawiadamia serwer, że chce pobrać dany plik i nie ma jeszcze żadnych bloków.
2. Serwer sprawdza do którego seenera/peera jest najkrótsza kolejka i na tej podstawie wybiera blok i hosta, od którego ten blok będzie pobierany. Ponadto inkrementuje wybranemu hostowi licznik.
3. Klient kontaktuje się z peerem/seederem wskazanym przez serwer, po czym następuje pobieranie bloku A.
4. Klient ma jeszcze jeden wolny socket, więc wysyła do serwera prośbę o wyznaczenie kolejnego hosta, od którego będą pobierane brakujące bloki.
5. Serwer wykonuje taką samą operację jak w pkt. 2 (nie uwzględniając już bloku A).
6. Klient podobnie jak w pkt. 3 kontaktuje się z innym klientem, po czym następuje pobieranie bloku B.
7. Blok A skończył się pobierać, więc klient informuje o tym serwer, który aktualizuje licznik hosta, od którego został pobrany blok A. Klient prosi o kolejny blok pliku (nie uwzględniając bloków A i B).
8. Serwer wskazuje klientowi hosta, od którego ma pobrać blok C.
9. Klient kontaktuje się z #4 w sprawie bloku C, po czym następuje jego pobieranie.
10. Blok B skończył się pobierać, więc klient informuje o tym serwer, który aktualizuje licznik hosta, od którego został pobrany blok B. Klient widzi, że wszystkie bloki są już pobrane lub w trakcie pobierania, więc nie prosi serwera o kolejny blok.
11. Blok C skończył się pobierać, więc klient informuje o tym serwer, który aktualizuje licznik hosta, od którego został pobrany blok C. Klient widzi, że wszystkie bloki zostały już pobrane, więc składa plik w całość, a procedura pobrania pliku kończy się.



Realizacja

Pliki

Plik podzielony na logiczne bloki o pewnym stałym rozmiarze. Blok jest najmniejszą jednostką zarządzaną przez serwer i klientów. Każdy plik jest ściągany w blokach, a nie całości.

W trakcie ściągania bloki plików są trzymane w oddzielnym katalogu (Blocks). Po pobraniu wszystkich bloków pliku, następuje jego scalenie i przeniesienie do innego katalogu (Files).

Serwer

Serwer trzyma listę deskryptorów plików i listę klientów.

Deskryptor pliku składa się z nazwy, rozmiaru pliku, listy bloków oraz ich haszy.

Do każdego bloku przypisana jest lista wskaźników na klientów, którzy posiadają dany blok.

Deskryptor klienta zawiera takie informacje jak:

- Czy klient jest aktywny
- Adres i port
- Licznik pobierających i czekających dopiero na pobranie
- Stan klienta (czy udostępnia, pobiera, czy to i to)

Jeśli nieznanomy klient kontaktuje się z serwerem to jest on dopisywany do listy klientów.

Klient

Klient komunikuje się z peerami/seederami za pomocą adresów otrzymanych od serwera, którego adres jest stały i znany przez wszystkich klientów. Może być utrzymywanych **x** połączeń wychodzących i **y** połączeń przychodzących oraz po jednym na połączenia z serwerem.

Istnieje lokalny katalog ze ściągniętymi już plikami, oraz katalog zawierający bloki, które jeszcze nie zostały pobrane w całości. Bloki są składowane w osobnych plikach pod nazwą: nazwa.pliku_numer.bloku (np.: example_8). Po pobraniu wszystkich, pliki te są sklejane w jeden docelowy.

Obsługa błędów

Serwer nie odpowiada klientowi

Zwrócenie błędu informującego użytkownika o braku możliwości połączenia się z serwerem.

Klient nie odpowiada serwerowi

Klient zostaje przez serwer usunięty z listy aktywnych hostów.

Klient (peer/seeder) nie odpowiada innemu klientowi (leech/peer)

Po minięciu limitu czasu oczekiwania na odpowiedź, klient ponownie wysyła do serwera prośbę o wybór hosta, od którego ma nastąpić pobranie bloku.

Klient (leech/peer) nie odpowiada innemu klientowi (peer/seeder) po zwolnieniu się kolejki

Po minięciu limitu czasu oczekiwania na odpowiedź, seeder wybiera kolejną osobę z kolejki (jeśli jeszcze ktoś czeka w kolejce).

Blok nie zostaje pobrany w całości

Blok zostaje usunięty z pamięci, a klient ponownie wysyła do serwera prośbę o wybór hosta, od którego ma nastąpić pobranie bloku.

Pobrany blok nie jest poprawny (hash na serwerze jest inny)

Taki blok zostaje usunięty, i następuje próba jego ponownego pobrania. Po n-tej próbie pobrania, serwer powinien usunąć hosta z listy posiadających bloki pliku.

Żaden z aktywnych peerów/seederów nie posiada danego bloku pliku

Jeśli jest to ostatni potrzebny blok, klient powinien oczekiwać na powrót hosta, który mu go dostarczy.

Klient nie posiada żądanego przez innego klienta bloku pliku

Serwer aktualizuje listę hostów udostępniającą dany plik (usuwa z niej hosta, który nie posiada już pożądanego pliku).

Podział zadań

Mateusz Bochenek

- Klient (udostępnianie plików)

Krzysztof Najda

- Klient (komunikacja z serwerem w celu pobrania pliku)

Karol Rzepka

- Klient (komunikacja z seederem)

Piotr Zawadka

- Serwer