W trakcie projektowania aplikacji internetowych nie trzeba się martwić o poszczególne gniazda TCP czy UDP, ponieważ zarządza nimi przeglądarka internetowa. Dodatkowo stos sieciowy zarządza ograniczeniem liczby połączeń, formatowaniem żądań, izolowaniem poszczególnych aplikacji między sobą oraz obsługę serwera proxy . Ale najważniejsze jest aby mimo wszystko nie zapominać o tym, że wydajność protokołów TCP, http czy sieci mobilnych w dużej mierze wpływa na jakoś działania tworzonej aplikacji.

Nowoczesne przeglądarki internetowe są tworzone z myślą o szybkim i bezbłędnym dostarczaniu treści poprzez internet. Zawierają w sobie ogromną liczbę warstw optymalizacji obsługi stron www. Popularne programy do odwiedzania stron w internecie zawierają moduły zarządzające procesami, dbające o bezpieczeństwo w sieci, optymalizujące pamięć podręczną, renderowanie grafiki, maszyny wirtualne Javy, obsługę audio-wideo i wiele innych. To prawie systemy operacyjne dla sieci.

Ogólna wydajność przeglądarki i dbałość o jej najbardziej aktualną wersję ma wielki wpływ na ogólną wydajność i szybkość uruchamiania się aplikacji internetowych. Dlatego w projekcie, który przygotowano(PRZYPIS-odwołanie do projektu) można sprawdzić nie tylko obsługę aktualnych rozwiązań technologicznych takich jak HTML 5, CSS 3. Dzięki zaimplementowanej w projekcie bibliotece „modernizr”, można określić poprawność takich opcji jak caching strony, obsługę video dla HTML 5 etc. Nawet najlepsza optymalizacja serwisu nie będzie miała wpływu na wydajność jeśli użytkownik końcowy w dalszym ciągu posługuje się przeglądarką Internet Explorer 6 zaimplementowanej w Windows XP. W przeciwieństwie do IE, Google Chrome „uczy się”(PRZYPIS) działać coraz szybciej w trakcie jego obsługi przez użytkownika. Przeglądarka „zapamiętuje” kolejność przeglądania stron, najczęstsze wyszukiwania, sposób zadawania zapytań do wyszukiwarki. Wszystko ma na celu optymalizację działania poprzez uprzedzanie działań użytkownika tak aby kolejne „zadania” wykonywały się szybciej, niż miałoby to nastąpić bez tej optymalizacji. Chrome wstępnie rozpoznaje nazwy za pomocą DNS, wcześniej nawiązuje połączenie TCP, aby otrzymać wstępny wygląd strony.

Stos obsługi sieci we współczesnych przeglądarkach jest nie tylko menedżerem gniazd komunikacji, to wręcz platforma do ich obsługi, która posiada własne kryteria optymalizacji, interfejsy API, wtyczki, dodatki i usługi. Każdy producent tych aplikacji uważa swoje parametry ustawień za najbardziej wydajne, to właśnie dlatego do wyboru jest pokaźna gama takich programów do korzystania z internetu jak, Chrome, Opera. Internet Explorer, Safari, Firefox czy Chromium.

Cykl życia pojedynczych gniazd komunikacyjnych nie jest zarządzany przez aplikacje internetowe uruchomione przez przeglądarkę. Dzięki temu krytyczne mechanizmy optymalizacji wydajności są zautomatyzowane. Np. ponowne wykorzystanie gniazd, nadawanie priorytetów żądaniom, negocjacje protokołów, ograniczenie liczby powiązań, późne wiązania itd. Założeniem przeglądarek jest oddzielenie zarządzania cyklem życia żądań od zarządzania gniazdami komunikacyjnymi.

Gniazda są grupowane w pule według źródeł, każda z nich pula określa własne zasady bezpieczeństwa i wymagania dotyczące liczby połączeń. Żądanie, które zostanie zawieszone, jest ustawiane w kolejne i zostanie mu nadany priorytet. Następnie każde żądanie zostanie powiązane z gniazdem w puli. Gniazdo może zostać użyte ponownie przez wiele żądań, jeśli sam serwer nie zakończy połączenia. Umieszczanie gniazd w puli i ponowne użycie połączenia TCP odbywa się automatycznie, dzięki czemu zyskuje wydajność aplikacji. Przeglądarka może optymalizować pulę i zamykać nieużywane gniazda. Obsługiwać żądania w kolejce według priorytetów. Aktywnie otwierać gniazda i uprzedzać żądania. Minimalizować opóźnienie i zwiększać prędkość transmisji poprzez ponownie wykorzystanie gniazd komunikacyjnych.

Usługi takie jak zarządzanie gniazdami komunikacyjnymi, połączeniami, pamięcią podręczną, bezpieczeństwo, obsługa żądań i odpowiedzi są realizowane przez odpowiednie typy połączeń. HTTP, WebRTC, XMLHttpRequest, czy Server-Sent Events wykorzystują odpowiednio wszystkie bądź tylko niektóre z wymienionych usług. W internecie jest tyle różnych stron czy aplikacji ilu jest ich twórców, dlatego też nie ma jednego najlepszego protokołu czy interfejsu API w pełni wydajnie działającego dla wszystkich typów połączeń czy usług. W zależności od potrzeb aplikacji może ona korzystać z jednej lub kilku różnych protokołów transportowych, opóźnień komunikatów, w różny sposób wykorzystywać pamięć podręczną przeglądarki. Poniższa tabela przedstawia porównanie funkcjonalności protokołów XHR, SSE oraz WebSocket. Wybór odpowiedniego może stanowić „drugie życie” dla wolno działających i niewydajnych aplikacji, które są źle odbierane przez użytkowników.

|  |  |  |  |
| --- | --- | --- | --- |
|  | XMLHttpRequest | Server-Sent Events | WebSocket |
| Strumieniowanie żądań | tak | nie | tak |
| Strumieniowanie odpowiedzi | ograniczone | tak | tak |
| Mechanizm ramkowania | tak | nie (base64) | ramkowanie binarne |
| Binarna transmisja danych | tak | tak | tak |
| Kompresja | tak | tak | ograniczona |
| Aplikacyjny protokół transportowy | HTTP | HTTP | WebSocket |
| Sieciowy protokół transportowy | TCP | TCP | TCP |

Tabela 123 Wysokopoziomowe funkcjonalności protokołów

Przeglądarki internetowe po wywołaniu adresu jakiejś strony, sprawdzają czy zasób nie jest przechowywany w ich pamięci podręcznej, jeżeli tak to zwracana jest zawartość z kopii zapisanej lokalnie. W przeciwnym przypadku, wysyłane jest żądanie do sieci, a po otrzymaniu odpowiedzi jest ona umieszczana w cache-u przeglądarki, tak aby była dostępna w przypadku kolejnego żądania i dostęp do niej niemal natychmiastowy. Przeglądarki internetowe automatycznie wykonują pewne czynności:

* Przetwarzają zlecenia dostępu do każdego zasobu
* Zarządzają wielkością pamięci podręcznej i usuwają jej zasoby
* Nadają ważność nieaktualnym zasobom

Aby przeglądarka po stronie klienta mogła prawidłowo zarządzać pamięcią, należy zadbać o to aby serwer zwracał prawidłowe zlecenia dostępu do pamięci podręcznej. Odpowiednie nagłówki takie jak ETag, Last-Modified oraz Cache-Control odpowiednio przygotowane pomagają wydajnie obsługiwać stronę www.

https://www.igvita.com/posa/high-performance-networking-in-google-chrome/