

網頁設計 HW6-http

資工系 B0929043 莊泓德

HTTP 1.0 是第一個在通訊中指定版本號的 HTTP 協議版本，至今仍被廣泛採用，特別是在代理伺服器中。

HTTP/1.1 是當前版本，持久連線（Connection : Keep-Alive）被預設採用，並能很好地配合代理伺服器工作，還支援以管道方式同時傳送多個請求，以便降低線路負載，提高傳輸速度。

HTTP/2.0 在 HTTP 1.x 的基礎上，大幅度的提高了 web 效能，減少了網路延遲。

HTTP1.0 和 1.1 在之後很長的一段時間內會一直並存，這是由於網路基礎設施更新緩慢所決定的。

http1.0 協議頭裡可以設定 Connection:Keep-Alive 在 Http 首部裡設定 Keep-Alive 可以在一定時間內複用連線，具體複用時間的長短可以由伺服器控制，一般在 15s 左右。一段時間內的連線複用對 PC 端瀏覽器的體驗幫助很大，因為大部分的請求集中在一小段時間以內。但對移動 app 來說，成效不大，app 端的請求比較分散且時間跨度相對較大。所以移動端 app 一般會從應用層尋求其它解決方案。

在 http 1.1 和 http 2.0 中： Connection 的預設值就是 Keep-Alive，如果要關閉連線複用需要顯式的設定 Connection:Close。如果 client 使用 http1.1 / 2.0 協議，但又不希望使用長連結，則需要在 header 中指明 connection 的值為 close；如果 server 方也不想支援長連結，則在 response 中也需要明確說明 connection 的值為 close。不論 request 還是 response 的 header 中包含了值為 close 的 connection，都表明當前正在使用的 tcp 連結在當天請求處理完畢後會被斷掉。以後 client 再進行新的請求時就必須建立新的 tcp 連結了。

HTTP1.1 和 HTTP2.0 的最大優勢就是對於頭部阻塞問題的解決。

HTTP1.1 的解決方案 - http pipelining：和圖一相比最大的差別是，請求 2，3 不用等請求 1 的 response 返回之後才發出，而是幾乎在同一時間把 request 發向了伺服器。2，3 及所有後續共用該連線的請求節約了等待的時間，極大的降低了整體延遲。

Http 2.0 的解決方案：

多路複用（multiplexing） multiplexing 允許同時通過單一的 HTTP/2 連線發起多重的請求-響應訊息。在 HTTP/1.1 協議中瀏覽器客戶端在同一時間，針對同一域名下的請求有一定數量限制，超過限制數目的請求會被阻塞。這也是為何一些站點會有多個靜態資源 CDN 域名的原因之一，拿 Twitter 為例，http://twimg.com，目的就是變相的解決瀏覽器針對同一域名的請求限制阻塞問題。而 HTTP/2 的多路複用(Multiplexing) 則允許同時通過單一的 HTTP/2 連線發起多重的請求-響應訊息。因此 HTTP/2 可以很容易的去實現多流並行而不用依賴建立多個 TCP 連線，HTTP/2 把 HTTP 協議通訊的基本單位縮小為一個一個的幀，這些幀對應著邏輯流中的訊息。並行地在同一個 TCP 連線上雙向交換訊息。

首部壓縮 (Header Compression) HTTP/1.1 並不支援 HTTP 首部壓縮，為此 SPDY 和 HTTP/2 應運而生，SPDY 使用的是通用的 DEFLATE 演算法，而 HTTP/2 則使用了專門為首部壓縮而設計的 HPACK 演算法。

服務端推送 (Server Push)：伺服器推送是一種在客戶端請求之前傳送資料的機制。在 HTTP/2 中，伺服器可以對客戶端的一個請求傳送多個響應。Server Push 讓 HTTP1.x 時代使用內嵌資源的優化手段變得沒有意義；如果一個請求是由你的主頁發起的，伺服器很可能會響應主頁內容、logo 以及樣式表，因為它知道客戶端會用到這些東西。這相當於在一個 HTML 文件內集合了所有的資源。不過與之相比，伺服器推送還有一個很大的優勢：可以快取，也讓在遵循同源的情況下，不同頁面之間可以共享快取資源成為可能。