

## hw6-http

B0929059 陳緯宸

### Http 1.0

HTTP 協議老的標準是 HTTP/1.0，為了提高系統的效率，HTTP 1.0 規定瀏覽器與伺服器只保持短暫的連線，瀏覽器的每次請求都需要與伺服器建立一個 TCP 連線，伺服器完成請求處理後立即斷開 TCP 連線，伺服器不跟蹤每個客戶也不記錄過去的請求。

這樣做的缺點是由於每次客戶端向伺服器請求時都要重新建立 TCP 連線，當網頁中包含有外部的靜態資源的載入，如：圖片，css 和 JS 檔案等，客戶端還要根據這些資源的 URL 重新與伺服器進行 TCP 的連線，即使影像檔案都很小，但是客戶端和伺服器端每次建立和關閉連線卻是一個相對比較費時的過程，並且會嚴重影響客戶機和伺服器的效能。

因此，Http1.0 最明顯的缺點之一就是連線無法複用，客戶端是依據域名來向伺服器建立連線，一般 PC 端瀏覽器會針對單個域名的伺服器同時建立 6~8 個連線，手機端的連線數則一般控制在 4~6 個。顯然連線數並不是越多越好，資源開銷和整體延遲都會隨之增大。連線無法複用會導致每次請求都經歷三次握手和慢啟動。三次握手在高延遲的場景下影響較明顯，慢啟動則對檔案類大請求影響較大。

Http1.0 的另外一個問題則是 head of line blocking: 會導致頻寬無法被充分利用，以及後續健康請求被阻塞。假設有 5 個請求同時發出，對於 HTTP1.0 的實現，在第一個請求沒有收到回覆之前，後續從應用層發出的請求只能排隊，請求 2, 3, 4, 5 只能等請求 1 的響應回來之後才能逐個發出。網路通暢的時候影響不大，一旦請求 1 的 request 因為什麼原因沒有抵達伺服器，或者伺服器響應因為網路阻塞沒有及時返回，影響的就是所有後續請求，問題就變得比較嚴重了。

所以 Http 1.1 和 Http 2.0 主要就是針對這兩個問題提出瞭解決方案和辦法：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。

缺點: pipelining 只能適用於 http1.1，一般來說，支援 http1.1 的 server 都要求支援 pipelining。只有幂等的請求（GET，HEAD）能使用 pipelining，非幂等請求比如 POST 不能使用，因為請求之間可能會存在先後依賴關係。head of line blocking 並沒有完全得到解決，server 的 response 還是要求依次返回，遵循 FIFO(first in first out)原則。絕大部分的 http 代理伺服器不支援 pipelining。和不支援 pipelining 的老伺服器協商有問題。可能會導致新的 Front of queue blocking 問題。正是因為有這麼多的問題，各大瀏覽器廠商要麼是根本就不支援 pipelining，要麼就是預設關掉了 pipelining 機制，而且啟用的條件十分苛刻。

#### Http 2.0 的解決方案

多路複用（multiplexing），請求優先順序（request prioritization），首部壓縮（header packet），伺服器推送（server push）