

網頁程式設計 HW6-HTTP

資工二甲 B0929056 陳冠宇

Http (HyperText Transfer Protocol) 稱為超文本傳輸協定，是一種用於分佈式、協作式和超媒體訊息系統的應用層協定。HTTP 是全球資訊網的數據通訊的基礎。設計 HTTP 最初的目的是為了提供一種發布和接收 HTML 頁面的方法。透過 HTTP 或者 HTTPS 協定請求的資源由統一資源識別碼 (Uniform Resource Identifiers) 來標識。

Http 已經演化出了很多版本，它們中的大部分都是向下相容的。在 RFC 2145 中描述了 HTTP 版本號的用法。客戶端在請求的開始告訴伺服器它採用的協定版本號，而後者則在回應中採用相同或者更早的協定版本。

HTTP/1.0 是 Http 協議的老式標準，HTTP/1.0 規定 Browser 與 Server 只保持短暫的連線，瀏覽器的每次 Request 都需要與伺服器建立一個 TCP 的三向交握 (Three-way Handshake) 以及慢啟動 (Slow Start) 的連線方式，Server 完成 Request 處理後立即斷開 TCP 連線，Server 不跟蹤每個客戶也不記錄過去的 Request。但是，這也造成了一些效能上的缺陷。HTTP/1.0 被抱怨最多的就是連線無法複用，以及 Head Of Line Blocking (HOLB) 這兩個問題。連線無法複用會導致每次 Request 都經歷三向交握和慢啟動。三向交握在高延遲的場景下影響較明顯，慢啟動則對大型檔案 Request 影響較大。HOLB 會導致頻寬無法被充分利用，以及後續健康的 Request 被阻塞。HOLB 會導致健康的 Request 會被不健康的 Request 影響，而且這種體驗的損耗受網路環境影響，出現隨機且難以監控。為了解決 HOLB 帶來的延遲，協議設計者設計了一種新的 Pipelining 機制。Pipelining 只能適用於 HTTP/1.1，而且由於使用苛刻，很多瀏覽器廠商並不支援。

HTTP/1.1 為了克服 HTTP/1.0 的這些缺陷，HTTP/1.1 支援持久連線 (Connection: keep-alive)，HTTP/1.1 的預設模式使用帶流水線的持久連

線，在一個 TCP 連線上可以傳送多個 HTTP 的 Request 和 Response，減少了建立和關閉連線的消耗和延遲。HTTP/1.1 還允許客戶端不用等待上一次 Request 結果返回，就可以發出下一次 Request，但 Server 端必須按照接收到客戶端 Request 的先後順序依次回送 Response，以保證客戶端能夠區分出每次 Request 的 Response 內容，這樣也顯著地減少了整個下載過程所需要的時間。HTTP/1.1 在繼承了 HTTP/1.0 優點的基礎上，也克服了 HTTP/1.0 的效能問題。HTTP/1.1 通過增加更多的 Request Header 和 Response Header 來改進和擴充 HTTP/1.0 的功能。

綜上所述，HTTP/1.1 相較於 HTTP/1.0 協議的區別主要體現在：

1. Cache 處理
2. 頻寬優化及網路連線的使用
3. 錯誤通知的管理
4. 訊息在網路中的傳送
5. 網際網路地址的維護
6. 安全性及完整性

常用的 Request 方式則有 GET、POST、HEAD、PUT、DELETE、TRACE、CONNECT、OPTIONS 等...

HTTP/1.1 狀態程式碼及其含義，狀態程式碼有三位數字組成，第一個數字定義了 Response 的類別，且有五種可能取值：

- 1XX：指示資訊，表示 Request 已接收，繼續處理
- 2XX：成功，表示 Request 已被成功接收、理解、接受
- 3XX：重定向，要完成 Request 必須進行更進一步的操作
- 4XX：客戶端錯誤，Request 有語法錯誤或 Request 無法實現
- 5XX：伺服器端錯誤，Server 未能實現合法的 Request

雖然 RFC 2616 中已經推薦了描述狀態的短語，例如 "200 OK"，"404 Not Found"，但是 WEB 開發者仍然能夠自行決定採用何種短語，用以顯示在地化的

狀態描述或者自訂訊息。

HTTP/2.0 是目前的版本，使用 <https://http2.akamai.com/demo> 便可看出 HTTP/2.0 比之前的協議在效能上有很大的提升。以下將總結 HTTP/2.0 協議的幾個特性。

1. 多路複用(Multiplexing)

多路複用允許同時通過單一的 HTTP/2.0 連線發起多重的 Request - Response 訊息。因此 HTTP/2.0 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/2.0 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。

2. 二進位制分幀

HTTP/2.0 在應用層(HTTP/2.0)和傳輸層(TCP、UDP)之間增加一個二進位制分幀層。在二進位制分幀層中，HTTP/2.0 會將所有傳輸的資訊分割為更小的訊息和幀(Frame)，並對它們採用二進位制格式的編碼，其中 HTTP/1.X 的 Header 資訊會被封裝到 HEADER Frame，而相應的 Request Body 則封裝到 DATA Frame 裡面。HTTP/2.0 通訊都在一個連線上完成，這個連線可以承載任意數量的雙向資料流。HTTP/2.0 通過讓所有資料流共用同一個連線，可以更有效地使用 TCP 連線，讓高頻寬也能真正的服務於 HTTP 的效能提升。這種單連線多資源的方式，減少服務端的連結壓力、記憶體佔用更少、連線吞吐量更大，而且由於 TCP 連線的減少而使網路擁塞狀況得以改善，同時慢啟動時間的減少，使擁塞和丟包恢復速度更快。

3. 首部壓縮(Header Compression)

HTTP/1.1 並不支援 HTTP Header 壓縮，為此 SPDY 和 HTTP/2.0 應運而生，SPDY 使用的是通用的 DEFLATE 演算法，而 HTTP/2.0 則使用了專門為 Header 壓

縮而設計的 HPACK 演算法。

4. 服務端推送 (Server Push)

服務端推送是一種在客戶端 Request 之前傳送資料的機制。在 HTTP/2.0 中，伺服器可以對客戶端的一個 Request 傳送多個 Response。服務端推送讓 HTTP/1.X 時代使用內嵌資源的優化手段變得沒有意義；如果一個 Request 是由使用者的主頁發起的，Server 很可能會 Response 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就是可以快取，也讓在遵循同源的情況下，不同頁面之間能夠共享快取資源。