Load balance on P4

小組成員：m10802136莊皓翔、m10802112陳棣文、m10915045施信宏、

m10915031鄭善謙、m10915080羅笠程

1. Introduction:

隨著科技的進步，資訊的傳播變得容易且快速，而在資訊爆炸的現今社會，為了因應網路巨量的資料需求，需要更有效率解決方案來讓傳輸速度提升，解決資料在網路上傳送所造成的壅塞問題，尤其在科技相關領域，效能更是關鍵，因此促使了負載平衡技術的發展，而目前在網際網路上能達到負載平衡的方式，透過軟體有 Haproxy、Linux Virtual Server (LVS)、Nginx、SDN，透過硬體有 F5 和 Array負載平衡器。

近年來，隨著新型的網路架構「軟體定義網路(SDN)」的被提出，徹底改變了網路傳輸與控制模式，軟體定義網路可以用來解決網路傳輸上的瓶頸，並能提升網路運作的彈性與效能，但傳統軟體定義網路的使用上，需要透過控制器將OpenFlow 的協定寫入 SDN 交換機內來對封包進行傳輸與分配，但傳統的 SDN 交換機並沒有辦法存取封包處理的規則與狀態，當控制器故障無法連接交換機時，就會使負載平衡功能無法運行，而 P4(Programming protocol-independent packet processors)的被提出了，解決了OpenFlow的不足，它的主旨在於 data plane可以編程化，它本身能不受封包標頭的支援限制且能夠自行定義新標頭，以及能夠存取封包處理的規則，還能存取封包處理的狀態，讓網路管理者方面可以更有效率的管理整個網路。

1. Purpose:

我們將透過 P4 設計出負載平衡機制來進行請求的分配，採用 Round Robin 和Hashing的方法，前者將客戶端請求按順序輪流分配到後端服務器上，以達到負載平衡的目的，後者通過雜湊函數計算得到的一個數值，用該數值對Host列表的大小進行運算，得到的結果便是客服端要訪問服務器的編號。

1. Related work:

本次實作前需要先了解最常被用來作為負載平衡的相關方法，以及SDN網路架構進行開發所需的相關工具。

1. Mininet

Mininet是一個網路模擬編排系統。可以自定義簡單的程式碼，讓它運行在單個Linux核心上且僅使用了輕量級的虛擬化即可讓單一系統像是擁有完整的網路環境，可以同時產生多個虛擬主機 / 交換機 / 控制器，並將其串接起來。Mininet上所創建的虛擬主機、交換機等網路設備是使用軟體去實做開發的，且在大多數情況下，它們的行為就類似於真實的硬體設備，所以在Mininet上所部屬的網絡拓樸，往往可以直接轉換於現實場景的網路環境，更能模擬在真實平台上所運行的程式碼與應用程式。

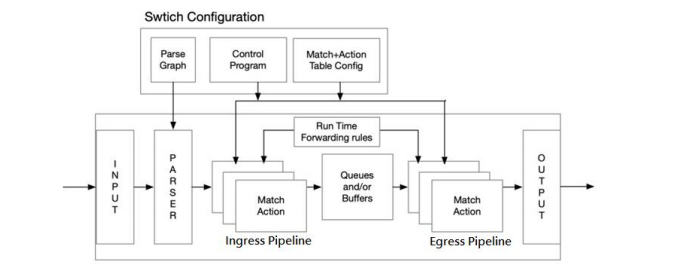
1. 軟體定義網路

軟體定義網路(Software Defined Network，SDN)的主要概念是將網路控制平面(Controller Plane)與轉發平面(Data Plane)的分離，將網路的管理權限交由控制層的控制器 (Controller) 軟體負責，採用集中控管的方式。其主要目的是希望能夠因應快速變化的網路環境，透過創新的想法與不同以往的網路架構，讓網路管理能夠更加靈活與容易操作。

1. P4(Programming protocol-independent packet processors)

P4語言是由數間大學（Stanford, Princeton）以及企業（Barefoor Netowrk, Intel, Google, Microsoft）於 2014 SIGCOMM 中提出，主要希望解決的問題是一般的 Switch，包含 SDN Switch 都只能夠處理一般的封包，例如 Ethernet、VLAN、IPv4 等等。

透過 P4，開發者能夠直接定義出一個 Switch 能夠處理的封包格式，舉例來說定義一個新的 Ethernet type，或是直接不用 Ethernet header，用自行定義的封包結構，另外就是他可以不被侷限現在特定的硬體之下執行，只需要有對應的編譯器就可以佈署。



上圖為P4 Switch抽象轉發模型圖，封包到達交換機後，先由parser 解析器來解析封包標頭提取出封包中的特定標頭字段，並執行 match Action，根據匹配的結果執行相關的動作，其中又分為Ingress和Egress，在Ingress封包可能會被轉發、複製又或者是丟棄，而在Egress處則執行對封包標頭的修改

1. 負載平衡機制

常見的負載平衡的機制如下：

* 1. 輪循（Round Robin）

輪詢法是負載平衡中最常用的演算法，它容易理解也容易實現。輪詢是指負載平衡器（load balancer）將客戶端請求按順序輪流分配到後端服務器上，以達到負載平衡的目的。

* 1. 加權輪詢（Weighted Round Robin）

簡單的輪詢法並不考慮後端機器的性能和負載差異。給性能高、負載低的機器配置較高的權重，讓其處理較多的請求；而性能低、負載高的機器，配置較低的權重，讓其處理較少的請求。而加權輪詢法可以解決這一個問題，它將請求順序且按照權重分派到後端服務器。

* 1. 最小連接數（Least Connections）

即使後端機器的性能和負載一樣，不同客戶端請求復雜度不一樣導致處理時間也不一樣。最小連接數法根據後端服務器當前的連接數情況，動態地選取其中積壓連接數最小的一台服務器來處理當前的請求，盡可能提高後端服務器的利用效率，合理的將請求分流到每一台服務器。

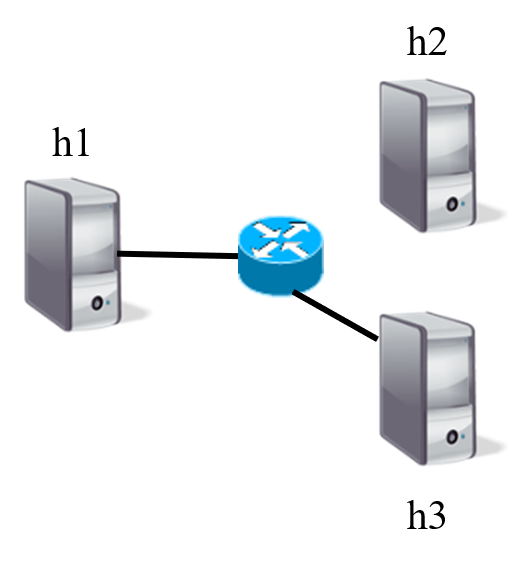
* 1. 隨機（Random）

隨機法簡單來說就是隨機選擇一台後端服務器進行請求的處理。由於每次服務器被挑中 的概率都一樣，客戶端的請求可以被均勻地分配到所有的後端服務器上。

* 1. 源地址雜湊（Source Hashing）

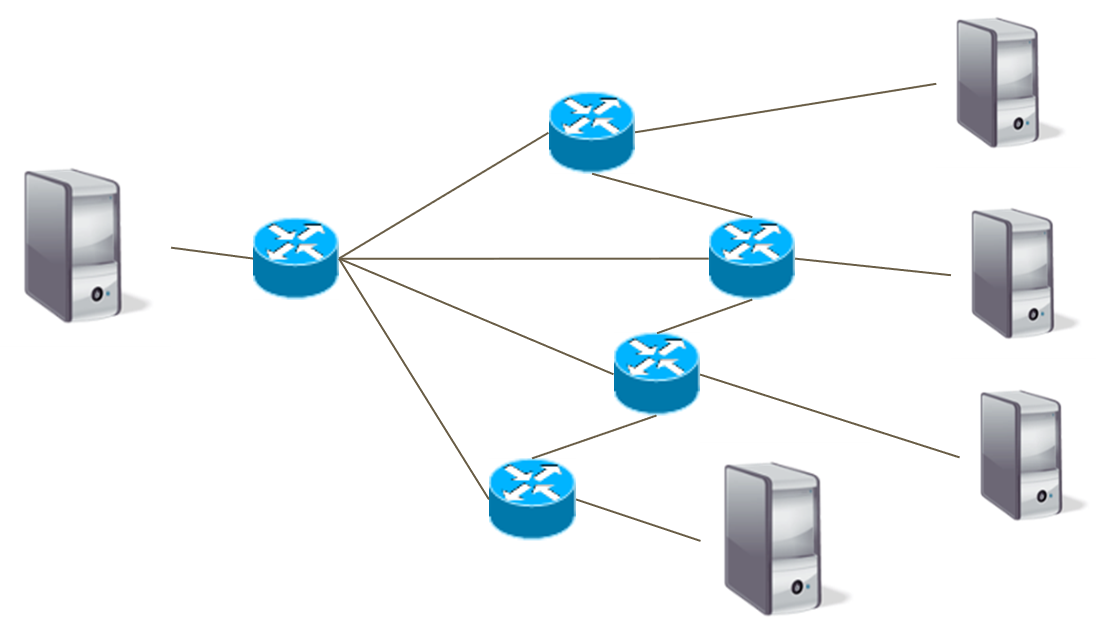
源地址雜湊的意思是根據獲取客戶端的 IP 地址，通過雜湊函數計算得到的一個數值，用該數值對服務器列表的大小進行取模運算，得到的結果便是客服端要訪問服務器的序號。採用源地址雜湊法進行負載均衡，同一 IP 地址的客戶端，當後端服務器列表不變時，它每次都會映射到同一台後端服務器進行訪問。

1. Methodology:
2. Round Robin



上圖為Round Robin的拓樸圖，Round Robin特別的地方為在egress 的部分加入register使用，紀錄了上一次是傳送至哪一台Host，下一次將讀取上一次的紀錄後傳送至另一臺Host。

1. Source Hashing



上圖為Source Hashing的拓樸圖，P4 switch透過雜湊函數計算得到的一個數值，用該數值對服務器列表的大小進行取模運算，得到的結果便是要訪問的Host序號。

1. Experiments result:
2. Round Robin

|  |
| --- |
|  |
| Host1送出200個封包，經過Round Robin後，Host2和Host3分別收到100個封包。 |

2. Source Hashing

|  |  |
| --- | --- |
|  | |
| Host1送出100個封包，經過雜湊函數計算後，依計算結果分別送入Host2、Host3、Host4、Host5 | |
|  |  |
| Wireshark中在Host2擷取到來自Host1的封包 | Wireshark中在Host3擷取到來自Host1的封包 |
|  |  |
| Wireshark中在Host4擷取到來自Host1的封包 | Wireshark中在Host5擷取到來自Host1的封包 |

1. Reference:
2. 基於P4 Switch的負載平衡器之設計與效能評估 ，作者 : 許世榮
3. 以負載平衡為考量之 SDN 網路路由研究 ， 作者 : 林詠棠
4. P4 official tutorial