

# 基於 P4 的網路負載均衡實作及探討

## An implementation and analysis of load balancing based on P4

指導教授：楊中平，專題成員：林建宇/李肇和，

開發工具：P4/PI/PTF/p4c/Behavioral Model (BMv2)/Mininet/Python，

測試環境：Linux Ubuntu 20.04

### 一、簡介：

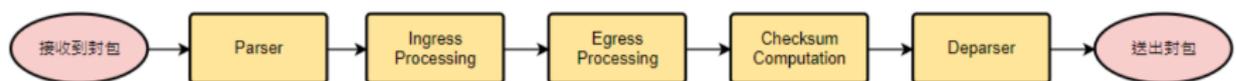
負載均衡是一種網路技術，運用於在多個伺服器或路徑之間分配網路流量，以提高資源利用率、減少響應時間並防止任何單一資源過載。這在大型數據中心、雲端計算環境和高流量網站中特別重要。

隨著全球化網路的蓬勃發展，當傳統靜態的網路拓樸無法再滿足不斷改變的網路需求，軟體定義網路（Software-Defined Networking, SDN）的出現顯得十分關鍵。SDN是一種網路管理方法，通過將網路的控制平面(control plane)和數據平面(data plane)分離，並將控制功能集中在一個或多個控制器上來進行集中管理和編寫。這使得網路更加靈活和可編寫，能夠快速適應變化的需求和條件。

SDN 透過 programmable switch 與 domain-specific language 來使 data plane 程式化。其中最廣泛使用的domain-specific language就是P4 (Programming Protocol-Independent Packet Processors)。P4是一種專為描述封包處理邏輯而設計的程序語言，使網路設備（如switch和router）可以靈活地配置和編寫，而不必依賴於硬件的固有功能。

以下為P4流程架構圖：

我們實作應用在web switch的軟體定義網路負載平衡技術，將網路流量合適的分

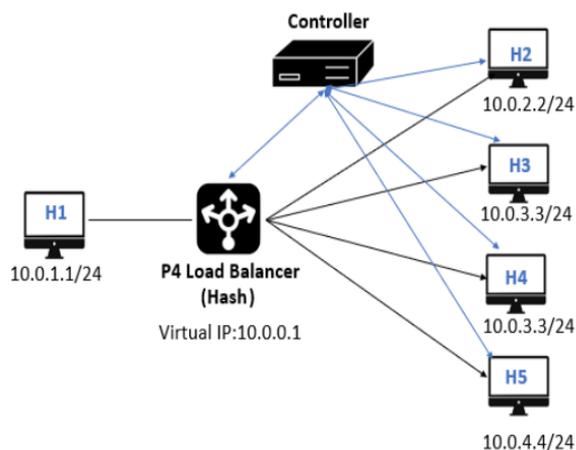


配到不同的伺服器中，提升主機或應用程式在流量高峰時的處理效能，以強化整體網路的穩定度；而當某個伺服器發生問題(需要維修更新、應用程式需要改版或負載過高)時，亦可將連線導入其他伺服器，待問題解決後再將連線重新導入，增加整體網路的靈活性。

## 二、測試結果：

Hash-based Load Balancing:

client傳送request給server時，會經過switch h1也就是右圖中的P4 load balancer，其作用是依據source IP, source port, destination IP, destination port, protocol(TCP or UDP)作為hash key。在產生hash key 結果後將封包分配給對應到的server。Controller會在每2秒進行一次health check來確認每台server的狀態是否正常，並適時做出調整。

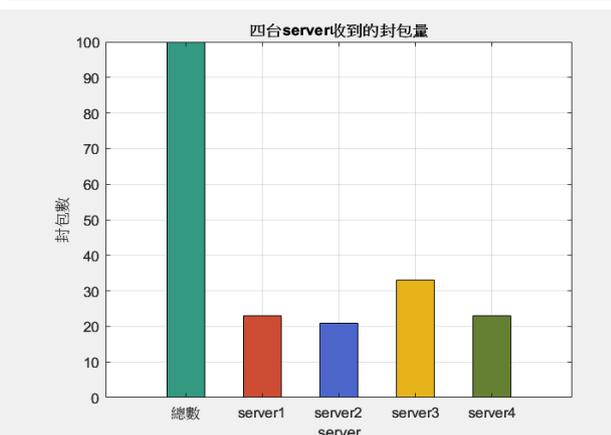
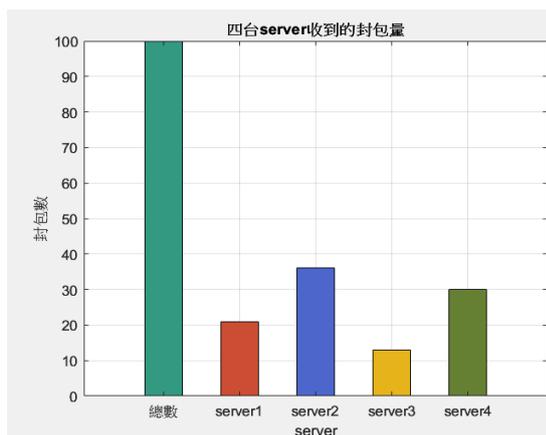


health check:藉由controller來向switch取得資訊，並依其判斷server是否有不健康的狀態。我們的設定是每台server收到的封包量，若是發生有兩台server發生封包量差距過大的情形，controller 會改變其中決定傳送封包的 flow entry，使收到最大量的server定義為unhealthy，並強迫switch 改變運送途徑，使封包傳送其他健康的 server。

我們送出100個封包來觀察各種結果

```
----- Reading tunnel counters -----
s1 MyIngress.Counter 0: 11 packets (770 bytes)
s1 MyIngress.Counter 1: 11 packets (770 bytes)
s1 MyIngress.Counter 2: 0 packets (0 bytes)
s1 MyIngress.Counter 3: 0 packets (0 bytes)
s1 MyIngress.Counter 4: 1 packets (70 bytes)
s1 MyIngress.Counter 5: 10 packets (700 bytes)
tiring_server:5 tiring_server_amount:10 easy_server:2 easy_server_amount:0

----- Reading tunnel counters -----
s1 MyIngress.Counter 0: 12 packets (840 bytes)
s1 MyIngress.Counter 1: 12 packets (840 bytes)
s1 MyIngress.Counter 2: 0 packets (0 bytes)
s1 MyIngress.Counter 3: 0 packets (0 bytes)
s1 MyIngress.Counter 4: 1 packets (70 bytes)
s1 MyIngress.Counter 5: 11 packets (770 bytes)
tiring_server:5 tiring_server_amount:11 easy_server:2 easy_server_amount:0
server 5 has too much loading!!
```



左下圖為無health check的情況，server接收的封包量不太平衡；

右下圖為有health check，可觀察到server接收的封包量平衡了許多。

### Reference

1. Load Balancing Implementation in P4  
[https://hackmd.io/@6VE0\\_f7yQw-TW\\_66oMIYow/r1A2H6OSj](https://hackmd.io/@6VE0_f7yQw-TW_66oMIYow/r1A2H6OSj)
2. Load Balancing Using P4 in Software-Defined Networks  
<https://jit.ndhu.edu.tw/article/viewFile/2400/2415>
3. P416 Language Specification  
<https://p4.org/p4-spec/docs/P4-16-v1.0.0-spec.html#sec-external-units>