網購超商快取？不不不~網路資料快取！

Implementing NetCache for Load Balancing Based on SDN with P4 switch

指導教授：張燕光; 專題成員：林宸哲 、古嘉雋

開發工具：p4-utils/python3/mininet/bmv2/p4c/p4runtime; 測試環境：Ubuntu 20.04

一、 簡介：

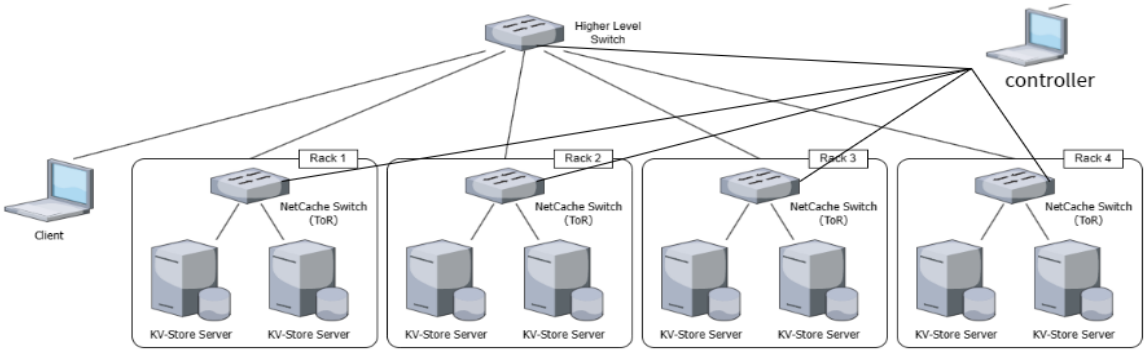
我們專題的動機是源於一篇論文(NetCache: Balancing Key-Value Stores with Fast In-Network Caching)，其作者想透過在可編程式交換器(P4 Switch)上實現 Key-Value 形式的資料快取功能，也就是 client 在查詢資料時須給出該資料的 Key 以得到對應的 value(資料本身)，而熱門資料會存在快取中，以解決有熱門資料的 servers 被造訪次數過多導致壅塞的流量不平衡的問題。

我們根據論文描述的 NetCache 概念，在 mininet 中模擬出了具有 netcache 功能的 P4 Switch 及 controller，switch 記錄下每份資料被查詢的頻率，並將被判斷為熱門資料的 Key 以及對應的 value 回報給 controller，controller 會將 Key 和查找 value 作為快取存進 switch 中，而 client 端在查詢資料時若資料的 key 有被快取到就能找到對應的 value，接著直接將資料返還給 client，而不需經過 switch 到 server 端，以此來降低熱門 server 的造訪次數。NetCache 也在顧及 Cache Coherence 的情況下有更新及刪除資料功能，也就是說 client 端若想要更新或刪除位於 server 端的某些資料時，若該資料也正位於 switch 快取之中，他也會被同步刪除或更新，以免發生快取資料與 server 資料不同的問題，此外若 switch 記憶體用量達一定標準則會以 LRU+LFU 方式刪除快取。

然而，論文中所採用的架構較為簡易，如圖一中框起來的部分，為單一 rack 架構，但在實際情況下通常會使用多個 racks 組成的複合架構。我們觀察到，就算快取功能有持續運作，不同 switch 內部的空間的利用不完全卻會讓快取效益降低，實際情況應會有某 switch 最先達到快取上限，而其他的 switch 卻仍有閒置的空間。因此，我們整合不同 swtich 中的儲存資源，令閒置的 switch 也能存入別人的 hot item。

我們在多個 racks 之上新增了一個 Higher Level Switch(HS)，透過各個 netcache switch 對 controller 回報各自快取用量，找出當前空間剩餘最多的 switch，將 hot item 存儲於其中，並在 HS 中新增一個 flow entry，讓新的查詢封包可以重新導向至實際存放該 item 的 switch 上。如此一來，switch 內的快取空間使用率便可以得到平衡。

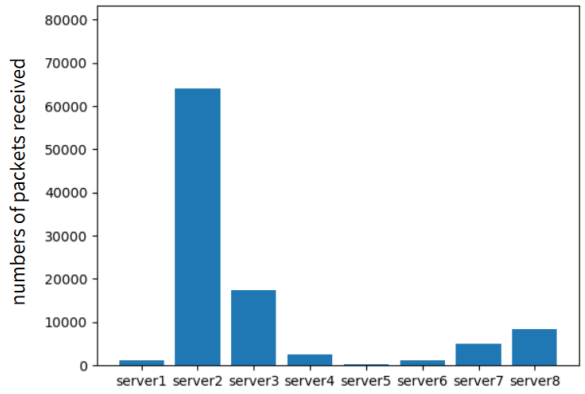
這是我們用於模擬測試的網路拓樸(圖一) ：



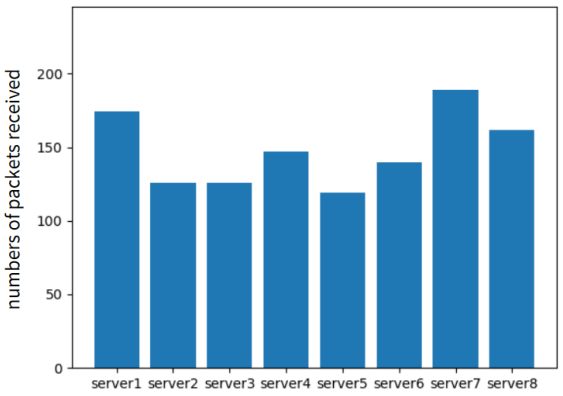
二、測試結果：

我們採用 zipf 機率模型生成了100000份查詢資料的封包，並且分別在有無 netcache 的情況下紀錄各個 server 端被查詢次數的情形 ，以及分別在有無 hot item 轉存的情況下紀錄各個 switch 隨時間增加的內部空間的使用率。

下面我們畫出在沒有(圖二)及有(圖三)netcache 的情況下各個 server 端被查詢的次數，X 軸為各 server 及其編號，Y 軸為收到的總封包數(被查詢次數)。可以看到沒有 netcache 的情況下 server 間的流量相當不均，次數也很多，但在有 netcache 時則大大減緩此現象。

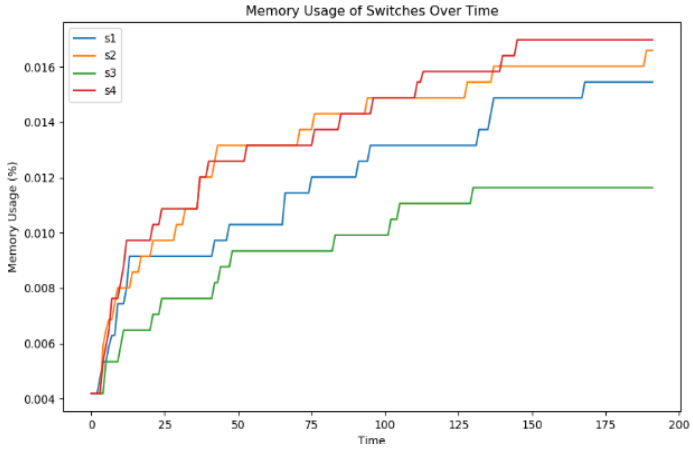


(圖二)

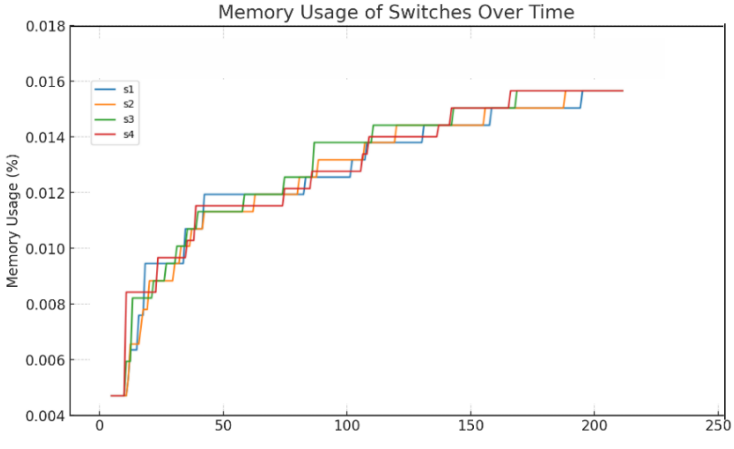


(圖三)

圖四、五為沒有以及有 hot item 轉存的情況下各個 switch 隨時間(X 軸)增加的內部空間的使用率(Y 軸)折線圖，可以看到在沒有 hot item 轉存時，某些 switch 的使用率會快速上升，而其餘的則上升緩慢，若快速上升的 switch 開始汰換自身 cache 時其他 switch 仍有閒置空間，會造成空間浪費，因此過多空間閒置應可以拿來多加利用，而有 hot item 轉存時所有 switch 內部空間使用率是平均增加的，如此一來可以避免閒置空間的產生並更有效的利用所有的空間。



圖四



圖五

三、參考資料：

[1] Xin Jin, Xiaozhou Li, Haoyu Zhang, Robert Soulé, Jeongkeun Lee, Nate Foster, Changhoon Kim, Ion Stoica, "NetCache: Balancing Key-Value Stores with Fast In-Network Caching", Proceedings of the 26th Symposium on Operating Systems Principles (SOSP), October 2017 https://doi.org/10.1145/3132747.3132764.

[2] The P4.org Architecture Working Group. 2022. P416 Portable Switch Architecture (PSA). https://staging.p4.org/p4-spec/docs/PSA-v1.2.pdf.