

Globally Distributed Content Delivery

一、動機

隨著網站越受歡迎、流量隨之而升，也就意味著網站很可能會遭遇 **flash crowd** 的問題，因瞬間過多的 request 致使不論是網站的前端、後端、頻寬或其他網路設備等基礎架構都將面臨挑戰，且還很可能因過高的回應時間令使用者對該網站產生負面印象，從而影響了網站利益。

此外，從單一位置所提供的網站服務，對其擴展性、可靠度、效能的發展都會有所限制，在缺乏彈性的情況下，影響小則網站不易壯大，影響大則網站容易 crash。

因此為了解決網站在**擴大服務中可能會產生的可靠度問題**，Akamai 提出了一套在世界各地多個地點建立代理來源伺服器(surrogate origin sever)並從離使用者最近的邊際網路(edge network)提供網站的快取內容的系統，除了可以**減少對原伺服器的負擔，也能提供使用者更快速的服務**。

相較於其他技術如:proxy，Akamai 因是從內容提供者的角度出發而設計，因此更能協助改善網路運作時的品質。Akamai 不僅可解決傳統內容提供者可靠性與擴充性的問題，尚得以**降低服務運作所需的設備與軟體門檻，令服務部署能更加快速與便利**。

二、議題

傳統解決網路流量高速成長的方式常見的有：叢集(clustering)、鏡像(mirroring)、路徑多宿(multihoming)，用以提升可靠度與可擴展性。雖這三種方式確實得以提供較大規模與較為穩定的服務，但這些方法並無法解決所也連線問題，且有可能反而會大幅增加原先的建置與維運成本。

- **Clustering**：透過許多電腦組成群組，協同合作完成計算任務，以期能達到高效率、高效能的目標。然而若欲將成千上萬台的伺服器組成叢集，在實際應用中是較為窒礙難行的。
- **Mirroring**：爲了提升節點的容錯率，利用異地備援的方式備份相同內容，以期能在節點發生異常狀況時能進行轉換。然而若欲布建至大規模網路世界中，因各節點所

有內容皆同步至最新狀態，而容易產生延遲、一致性問題。

- **Multihoming**：爲了提高連線時的容錯率，利用多個 ISP 以連接至伺服器，當其中有連線失敗時得以自動切換至其他備援線路。然而在特定網路協定下如:BGP 發生連線錯誤時，則會相當程度地影響重新彙整的連線速度，也就降低了使用者體驗。

面對網路的 flash crowd 問題，Akamai 雖提出了一個在世界各地多個地點建立代理來源伺服器(surrogate origin sever)，並從離使用者最近的邊際網路提供網站的快取內容之系統。然而，欲管理分散在世界各地的伺服器仍有許多技術上的難度，包含將使用者請求導向至適當的伺服器、運作時的錯誤失敗處理、監控並控制伺服器、更新或部署各地系統軟體等。

三、研發方向

✧ 自動化網路控制(Automatic Network Control)

- i. 當使用者發出 HTTP request 時，透過 Dynamic DNS 伺服器，利用 Akamai 網路拓模計算伺服器的 round-trip time、request service type、network status，決定封包欲轉交給哪個 content server。
- ii. 使用者發出 HTTP request 到 edge server 後，edge server 上若無快取內容，則會像其他 Akamai server 或 content provider 請求應該回應的資料內容，edge server 再將之回傳給使用者。

上述系統中，DNS server 與 edge server 皆相當重要，而爲了避免伺服器無法負荷，所有伺服器的狀態都會被及時監控記錄下來，並且在交付任務給伺服器時進行 **load-balance**。

✧ 監控伺服器狀態(Network Monitoring)

Akamai 系統會模擬使用者發出請求的狀況，下載網站物件並記錄其錯誤率與下載所需時間，以此兩項指標評估伺服器狀態是否正常。

而爲了避免網路壅塞，系統需要持續監控 state of service、service server 以及 service network。由 content server 頻繁地向 monitoring application 回報負載狀態，再由 monitoring application 生成負載報告交予 DNS server，DNS server 便可由此決定回傳的 content server IP，從而避免壅塞。

✧ 系統可靠度

當 Akamai 的 DNS 系統偵測到異常時，會指派另一新 IP 將使用者導向至其他 edge

server 以完成使用者請求。而若使用者端仍保存著先前快取的 DNS 解析結果，則會一次回傳多組 IP 供使用者選擇，或令其他伺服器偽裝為異常伺服器，使整個系統能順利繼續運作。

✧ 軟體部署與平台管理

在軟體升級方面會採取階段是進行，亦可減少因為更新而產生的 bug 影響範圍。惟因更新會有先後之差，故新舊版本間仍需能正常溝通以維持整體系統運行。

✧ 依目標內容提供各自解決方案

- i. **靜態內容**：包含 HTML、圖片、PDF 文件等，利用「Lifetime」標註該內容可在 content provider 中存活的時間，最小值為 0，最大值為無限期。
- ii. **動態內容**：將依據不同使用者而產生不同內容，因這些內容無法進行快取，故 Akamai 利用 ESI(Edge Side Includes technology)來解決。ESI 會將動態內容切割成許多獨立、可快取的片段，直至接收到不可被快取的片段，再進行組裝。Content provider 便只需要提供不可快取的內容即可，從而減少伺服器資源的消耗與資料量。
- iii. **影音串流**：Akamai 會將串流內容儲存於 entry-point server，並將這些內容複製到各地的 data center 存放。當 edge server 要求內容時，便可從最近的 data center 抓資料。Akamai 亦會監控 entry-point server 存放的影音，隨時清除無效或損毀的內容，以確保個檔案皆可正常存取。同時 Akamai 會監控雍塞的路徑，當 packet loss 過大時，會令 end point server 增加其他路徑以傳給 edge server。

四、 結論與未來

不論在需求、經濟效益、回應時間等方面，Akamai CDN 服務著實能為現今網路世界提供較為**快速可靠**且較**易擴展**的解決方案，不僅可**提升一般使用者的體驗**，同時也能**減輕內容提供者來源伺服器的負擔**。

在未來若需要壯大 web service，使用 CDN 技術可說是必要的，因此甚至可將 CDN 做為一種 Infrastructure，以維持使用者滿意度。另外在**資訊安全**方面，由於在世界各地許多的地點都擁有網站的複本，只要能夠攻破眾多伺服器的其中之一，便可能獲取隱私資料，安全便成為 CDN 的一大挑戰。或許能透過**自動化攻擊偵測**、**異常行為偵測**等方式，為 CDN 節點提供最佳的防護。