為台灣加油打氣專欄(241)高可靠度的雲端分散式系統

李家同

侯冠維

電腦和各種資訊系統已經成為我們日常生活中不可缺少的重要工具了，其應用數不勝數。舉例來說，許多人都有到ATM (自動櫃員機) 進行提款或轉帳的經驗，這部ATM當然是一部電腦，並且連接銀行內部的資訊系統，將我們的交易紀錄送到銀行去。我們生活中有許多活動也都依賴於電腦，例如買火車票、飛機票、演唱會門票、到醫院看病時掛號、看診、繳費、領藥等等。如果沒有可靠的電腦和資訊系統，這些服務可能將會錯誤百出。

圖一顯示了一個最簡單的資訊系統，透過一部伺服器來提供服務。這部伺服器可以是負責售票的伺服器，也可以是負責進行金融交易的伺服器，依我們想要提供的服務來決定。使用者要使用這些服務時，首先要連線到這部伺服器上。



圖一

然而，任何一架電腦都有可能會當機的，因為電腦程式是人寫出來的，也就有可能出錯。而電腦的硬體隨著使用年歲增加，損壞的機率也會增加。當伺服器突然當機或損壞時，這些重要的服務就可能發生中斷。更糟糕的是，假如儲存裝置損壞，資料可能會遺失。如果銀行的交易紀錄遺失，將是一場災難。因此大企業通常要購買非常高級的伺服器，其性能和可靠度要非常高，當然價格也會極為昂貴。

以下我們要介紹我國自行開發的高可靠度的雲端分散式系統，能夠確保當電腦當機或損壞時，可以降低資料遺失的風險，整個系統也可以迅速回復到正常狀態，請看圖二。



圖二

 當使用者要連上一個網站時，過去只有一部伺服器可以提供這個網站，因此只要這部伺服器損壞，這個網站就無法使用了。為了讓服務可以更加可靠，我們可以部署多個伺服器來提供這個網站。

我們以Google的網站為例，當使用者要連線到Google時，使用的是Google的網址：<https://www.google.com.tw/>，但是全世界有這麼多台電腦，電腦是如何知道Google的網站存放在哪裡呢？

當使用者在瀏覽器中輸入網址時，這個網址首先會被傳送到一個「域名伺服器」，如圖三所示。域名伺服器會幫我們找出這個網址所對應的伺服器IP位址，IP位址就如同電腦的門牌號碼，只要有了IP位址，就可以知道伺服器在哪裡。



圖三

域名伺服器中儲存了每一個網址所對應到的伺服器IP位址。我們要知道，一個網址可以對應到多台伺服器，也就是有很多台伺服器都可以提供這個網站。在圖三中，網址A就對應到兩台伺服器，域名伺服器會從清單中選出其中一台伺服器的IP位址，傳回給使用者。當使用者收到這個IP位址後，就可以用它和這台伺服器聯絡。

因此，只要有域名伺服器，我們就可以部署多台伺服器，讓它們都可以提供同一個網站，同一種服務。舉例來說，我們可以部署10台伺服器來販售演唱會門票，販售門票的網址是同一個，但是域名伺服器卻可以從10台伺服器中輪流選擇1台傳給使用者。如此一來，我們不用怕網站塞車。如果其中一台伺服器壞掉了，只要從域名伺服器中將它刪除，其它9台伺服器仍然可以正常服務。

 為了確保當其中一台伺服器損壞時，使用者的資料不會遺失，每一台伺服器 (Master) 都必須有至少一台伺服器可以作為備援伺服器 (Slave)，如圖四所示，伺服器2是伺服器1的備援。當使用者儲存了任何檔案到伺服器1上，伺服器1都必須將檔案完整的備份到伺服器2。當伺服器1當機或損壞，或者被駭客攻擊無法再正常運作時，伺服器2就可以立即接手伺服器1的工作，不會造成整個系統的崩潰。



圖四

 由於檔案備份時只有傳送到伺服器2，因此目前只有伺服器1和2有使用者1的檔案，假如使用者1下一次再連線時被分配到從伺服器3連線，那麼伺服器3上面就沒有這個檔案，這種情況要如何處理呢？請看圖五。



圖五

 當使用者1儲存了一個檔案到伺服器1時，由於伺服器2是伺服器1的備援伺服器 (Slave)，因此完整的檔案會被備份到伺服器2，同時伺服器1也會將檔案的索引傳送到伺服器3。檔案索引不是完整的檔案，而像是這個檔案的身分證，以及記錄了檔案目前儲存在哪裡。如此一來，下一次使用者1如果從伺服器3連線，伺服器3就可以根據檔案索引知道要向伺服器1取得這個檔案了。

我們以上介紹的是非常簡要的架構，當伺服器的數量從3台增加到10台甚至100台時，資料的同步與備份將變成一個比較複雜的問題，我們現在用一個例子來解釋。

假設現在有2台 master：A和B，A的slave是P，B的slave是Q和R，還有另外2台伺服器X和Y，所以共有7個伺服器。

現有d1，d2，…，d5共5筆檔案，假設d1和d2 放在A和P，d3、d4、d5放在B、Q和R。在A，P，B，Q和R都記錄了d1到d5放置的地點，X和Y沒有放d1到d5的任何檔案，但是也記錄了d1到d5放置的地點。因此 無論使用者被分配連上7個伺服器中的任何一個, 此系統都可以知道使用者要的資料放在哪一台伺服器中，並協助取得。

 一套可靠的資訊系統，除了要確保電腦當機或損壞時系統不會崩潰以外，更進一步是要確保資料不會有錯。舉例來說，假如我們去銀行匯款，牽涉到把錢從帳戶A轉移到帳戶B，這個過程當然是由銀行的伺服器來處理的。一次匯款至少包含了兩個步驟，請看圖六。



圖六

 這兩個步驟少了任何一個，都是嚴重的問題。假如步驟1有成功執行，但是此時電腦突然當機了，或是網路塞車，或是駭客入侵，導致步驟2無法執行，那麼一筆錢就憑空消失了。因此一個好的資訊系統，必須有能力確保這兩個步驟都能順利執行，假如其中一個步驟不幸失敗了，那麼另外一個步驟也不可以執行。也許各位會認為這聽起來很簡單，但是對於開發資訊系統的工程師來說，這卻是不容易做到的事。目前全世界對於這類高等級的應用系統，絕大多數都用了世界級的資料庫系統，資料庫系統會有解決上述確保資料完整性的功能作法。

 假設我們要執行兩個步驟A和B，A沒有執行的話不可執行B，但如果執行了A，B因當機而無法執行時，A的執行動作會自動還原為沒有執行，高級的資料庫系統有一個機制，那就是在A和B都執行完了以後會有一個確認的紀錄，如果沒有確認的紀錄，A和B的執行都不會生效，這樣做當然可以保證資料處理的正確性。

 我國自行開發的高可靠度的雲端分散式系統，已經能夠解決我們以上所提出的所有問題了。為了達成這個目的，工程師自行開發了進行伺服器之間同步與備份的演算法，並且也開發了一套資料庫系統。從上面的說明，如果要讓同一筆交易的所有交易執行步驟能完整的傳送給所有其他的分散式伺服器，就必須知道整筆交易會包含哪幾個交易執行步驟，使得所有的伺服器都能正確同步對應該有的所有資料和檔案。因此，自行開發的資料庫系統可以更有效地達成目的。相對的，如果利用坊間的資料庫系統，上述的交易紀錄資訊就可能必須在應用系統中自行處理，這幾乎是做不到的。

在過去，大多數好的資料庫系統只有國外的公司有能力做。因為要開發好的資料庫系統，必須非常懂得作業系統，必須有很好的邏輯思考的能力。如今我國的工程師也有能力開發資料庫系統，也因此有了分散式系統。這絕對不是一件容易的事，非常值得我們慶幸，我們應該給我國的軟體工程師鼓勵。

資訊系的學生們也應該對複雜的系統程式有興趣，不要只發展簡單的軟體。