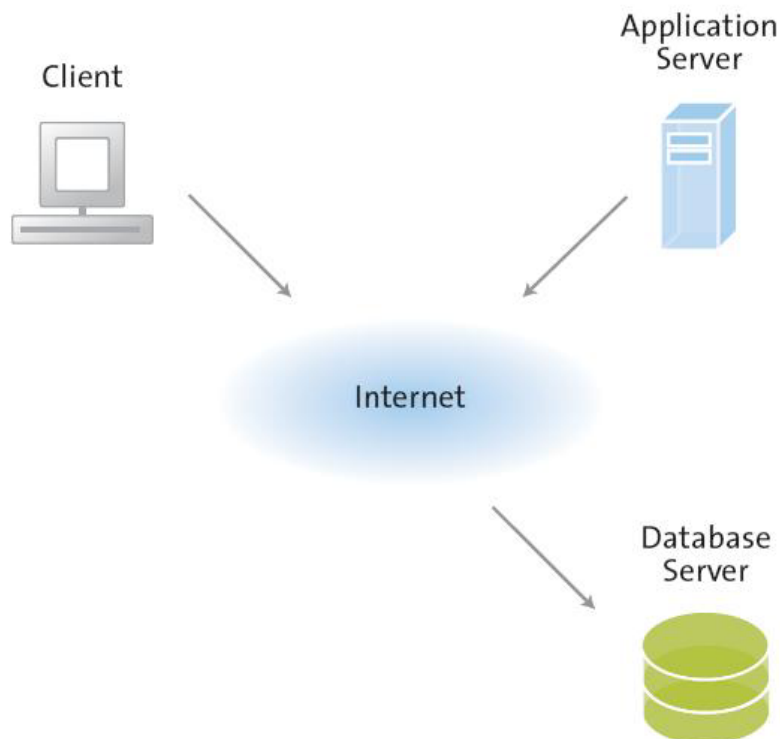


如何升級 MS SQL 至 Sybase ASE

資料庫在使用了一段時間後，也許會因為效能、功能、或其他軟硬體限制等問題而面臨必須轉換的困境。雖說關聯式資料庫(RDBMS)的原理都是相通的，但實務上各個不同的 DB 廠商還是有他們自己獨特的實作方式，因而造成資料庫移轉時許多相容性的問題。而在眾多的 RDBMS 中，相容性最高、相似度最大、轉換的成本最低的應該要算『本是同根生』的 MS SQL 與 Sybase ASE 了(皆源於 T-SQL 的標準)。但即使是這麼相似的兩個資料庫，在轉換過程中還是有很多要注意的地方，本文主要目的就是在探討如何進行從 MS SQL 轉換到 Sybase ASE 的通用流程。限於篇幅，文中無法將兩個資料庫的所有細節一一列舉，因此本文的目標讀者乃是對這兩個 DB 有基本認識的 DBA 或 Developer。您若想從基本學習起，請參考其他相關書籍或文章。

Migration 的主要步驟

除非是爲了學術研究的用途，否則資料庫很少獨立存在，它多半是與存取它的應用系統 (Application) 相互依存。因此當我們說資料庫移轉或升級，其實指的不只是資料庫，還包含了 AP 的部份。



上圖顯示出了一個包含用戶端、應用程式、與後端資料庫的完整應用系統架構。從這張圖就可以看出來，一個完整、嚴謹的 Migration，是應該要將上述三個部份都考量進去的。而 Sybase 官方建議的標準移轉流程就符合了這個原則，此流程主要包括：

- 資料的移轉(Data Migration)
- 應用程式的移轉(Application Migration)
- 移轉成果的驗證(Validation)
- 新系統的效能調優(Optimization)

以下內容將針對這些步驟一一詳述。

Data Migration

標題雖然只點到『Data』，其實這個步驟裡包含了：建立 Sybase 系統、建立與資料庫管理及安全相關的機制、最後才是資料的移轉。

首先你當然得先建立一個 Sybase 資料庫(否則資料要放哪兒)。而由於 MS SQL 與 Sybase ASE 存在著一些架構上的差異，你沒有辦法直接將 MS SQL 的資料庫及其物件的建立語法 (DDL)直接套用到 Sybase 上。

這裡舉兩個例子。

MS SQL 是透過所謂 Filegroup 的方式來分類、管理 Data File 以及 Log File；但在 Sybase 裡就沒有這種觀念。ASE 是透過 Device 的機制進行 Storage 的切割、分類與管理。因此之前以 Filegroup 的角度來看理所當然的一些設定，都必須改以 Device 的角度來重新檢討一遍。

另外一個例子則是關於空間增長的管理方式。在 MS SQL 裡，當預設的檔案空間用完後，系統可以自動的調增新的空間出來；但在 Sybase 裡就沒有這樣的機制，你要嘛手動管理這些空間，若一定要結合自動化管理，則必須額外設定 Threshold Action 來滿足你的需求。一個基於 DB 會自動增長的假設的 AP，轉到 Sybase 後若不做調整，將來一定會遇到問題。

除了這兩個與 Storage 的管理有關的例子外，其他在 Process、Query、Memory 等管理上，這兩個資料庫也有很多根本上的差異，若不能在建立 Sybase ASE 時列入考慮，則將來 AP 的運作幾乎能肯定絕對會出問題。

有了資料庫之後，接著就可把之前在 MS SQL 上所建立的一些管理或安全機制也搬到 Sybase 裡面來。而這也是不能直接抄襲就能了事的。舉個例子來說，很多在 MS SQL 裡面被整合成內建 Service 的功能，在 Sybase 裡多半被獨立成單一的伺服器，如 Monitor Server 或 Backup Server。又以安全控管為例，已經習慣於資料庫帳號與 Windows 系統帳號整合在一起的使用者，可別以為這一套在 Sybase 裡面也可依樣畫葫蘆，你必須在 Sybase 重新設定這些群組(Group)、或角色(Role)之類的安全機制才行。

一旦資料庫、連同裡面的各種管理與安全機制都建置完成，接著就可以進行實際資料的轉換了。可用來轉換資料的方式很多，例如：

- 透過 DB 自己提供的轉換機制：像是 MS DTS、Sybase CIS、Bulk Copy 指令等等。
- 透過 Third Party 廠商提供的相關工具：像是 PowerTranster、InfoMaker 等等。

各個工具都有他們的優缺點，有的可能受限於某些特殊資料型別無法轉換、有的則可能無法處理不同編碼方式所產生的資料。因此在挑選你要用的工具時，一定要先將這些工具的特性與限制搞清楚才行。(使用者若真有這方面的困擾或需求，可自行參閱 Sybase White Paper: MS SQL to Sybase ASE Migration Guide)

Application Migration

應用程式系統的移轉通常包含兩個部份：SQL Statement 的部份(本文所指的當然就是 T-SQL)、以及 Client AP 自己的 Language(例如 JDBC 或 ODBC)。

先來討論第一部份 – SQL Statement。這個範圍也蠻廣的，單獨的 SQL 指令、預存程序 (Stored Procedure)、觸發程式 (Trigger) 等等都是。雖然兩種資料庫都源於 T-SQL，差異不大，但還是要注意一些細節。例如，很多預存程序都會在程式內使用到一些系統變數或系統預存程序，而這些東西在兩個資料庫之間就有可能出現差異。像 MS SQL 有 sp_tableoption 而 Sybase 沒有；但 Sybase 的 sp_monitorconfig 微軟也沒有。因此習慣在 Stored Procedure 或 Trigger 中使用這些內建物件的人，在轉換後一定要詳細比對這些差異性才行。

至於 Client Language 的部份，除了上面舉的 JDBC 或 ODBC 例子外，像是 Web Application、Embedded SQL Application 等等也是。連結 MS SQL 與 Sybase 的連線字串語法我相信一定不一樣，這些若不跟著做出正確的轉換，就算內含的 SQL Statement 正確也沒有用。

Validation

這通常是整個轉換作業最重要、但也是最困難的部份。說它重要，是因為未經驗證怎能證明是否真的轉換成功？萬一看起來成功，但是將來跑出來的資料不是你要的怎麼辦？而說它困難，則是因為很難找到現成的工具來進行這個工作。實務上多半是 DBA 們自行建立檢測用的 Script 或 Procedure 來進行，難度當然較高。將來你要怎麼進行這個步驟，隨著環境與需求的不同，你會有不同的作法，本文無法預先猜測；但可以確定的是，一個嚴謹的驗證，至少要驗證下列項目：

- **User Interface Transaction**
- **Batch Processing**
- **Administration Procedures**
- **Disaster Recovery**
- **Application Performance Obtained**

October 07 M-Power eNews

本篇文章版權為倍力資訊股份有限公司所有，未經書面同意，嚴禁複製、轉載

Optimization

會不會發生一種狀況：資料庫的移轉雖然順利完成，但移轉後的效能卻變得出奇的差？當然有可能！因此在上面四個動作都完成後，最後我們要考慮的就是所謂最佳化調校的部份了。

之前在 MS SQL 所建立的效能調校機制，在轉到 Sybase 之後，若有相同的機制，當然要跟著轉過來；若沒有相同的機制，至少要找到相似的，這樣整體效能才不會差太多。舉個例子來說，蠻多人會將 Loading 最重的一些資料表 Mark 成所謂的 RAM resident，好讓這些資料表常駐於記憶體之中，達到效能提升的目的。可惜在 Sybase 裡面並沒有這樣的機制。但你還是可以透過 Sybase 的類似方法來達到同樣的效果，也就是善用他的『具名緩衝』(Named Cache)功能，使得特定的資料庫物件能夠擁有屬於他們自己的專屬緩衝記憶體，可說是 ASE 版的 RAM resident 機制。

結論

本文介紹了 Sybase 針對 MS SQL 轉換至 ASE 的官方標準流程、以及每個步驟的注意事項。雖然很多細節部分無法全部列舉，但只要能徹底掌握這些流程的內涵與精神，使用正確的工具，再加上對兩種資料庫都擁有基本的認識，相信要執行一次成功的轉換應該不是什麼難事才對。