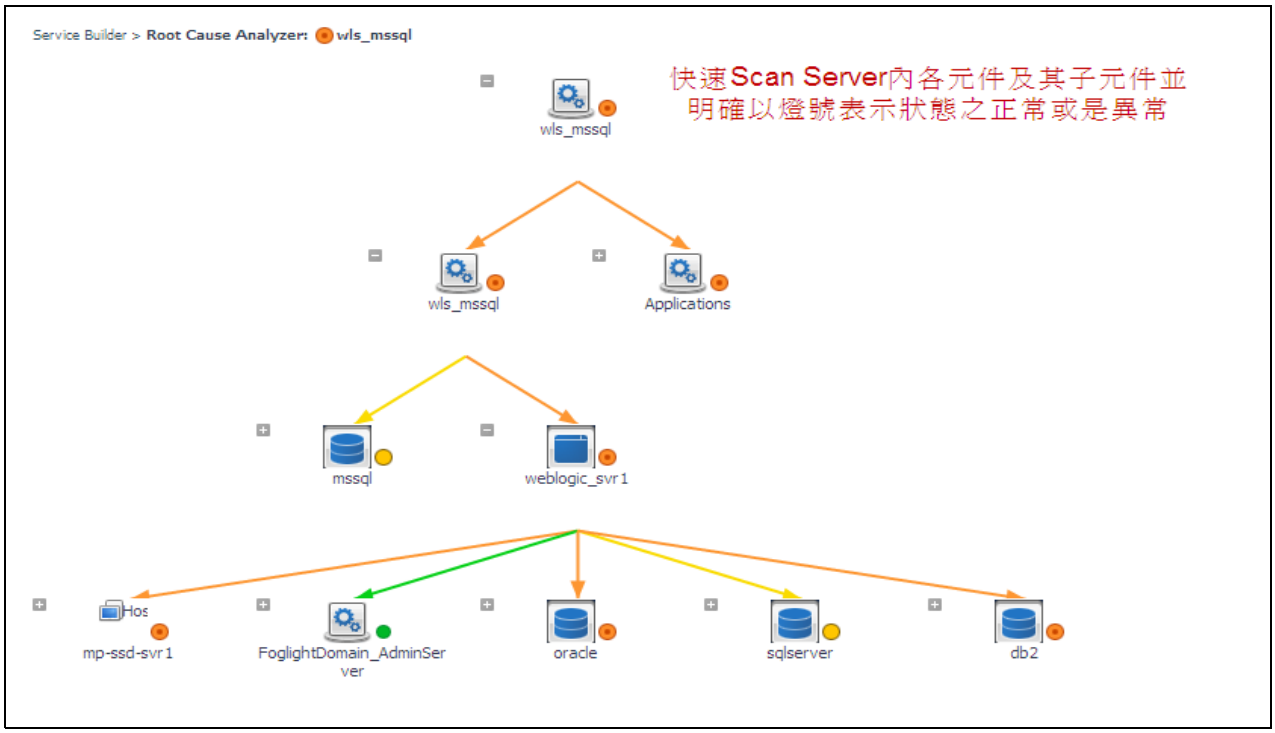
SSD 系統服務部 魏志強

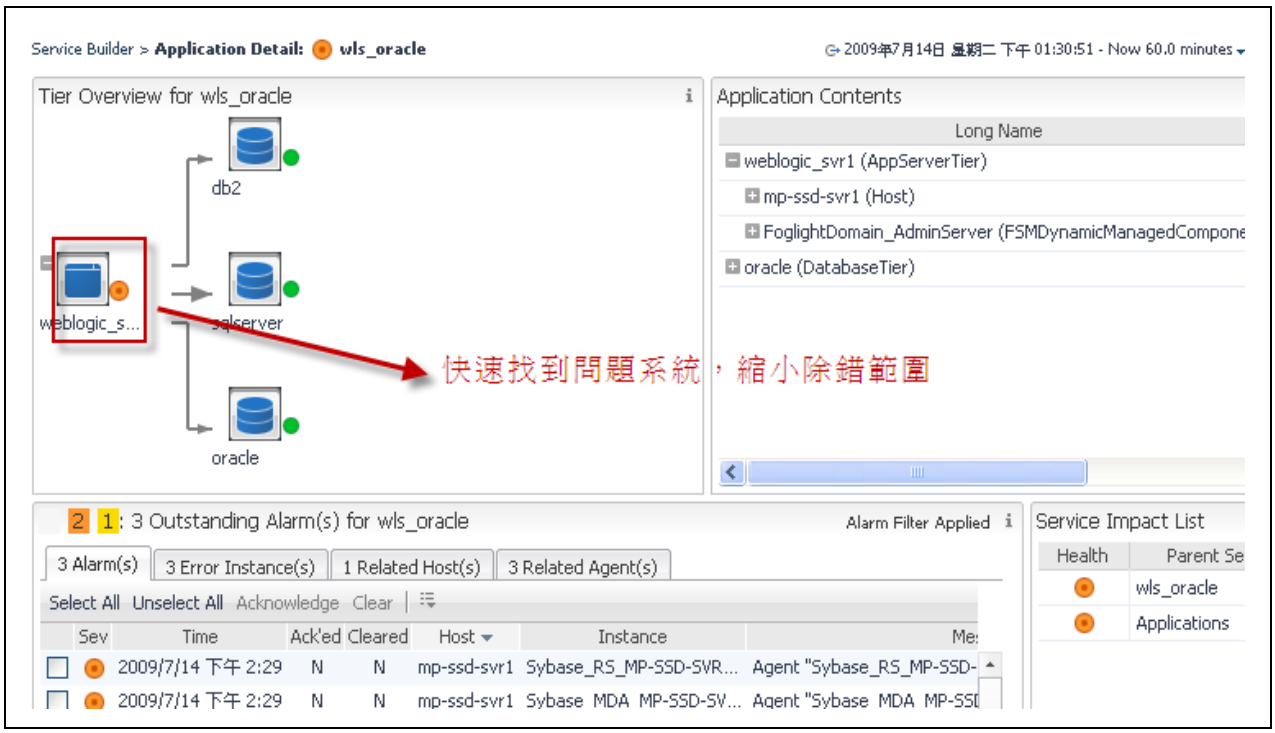
如何快速掌握系統問題的癥結

相信不少系統管理者均有遭遇過以下的情境，當一線客服人員告知 Application Server 的管理者，有使用者反應某個網頁的反應速度過慢或者整個變成白色的時，管理者檢查後亦不知問題出在哪個環節，因此認爲可能是後端資料庫有問題，而資料庫管理者檢查後亦認爲資料庫無問題，最後不知道問題何在，在不斷的重啓應用程式 Server 與資料庫無解後，只好找原廠支援，然而，這個狀況可能已經是半天或者一天過後，而等實際上廠商找到問題可能又過了半天，最後，整個 Service 停擺或無效率的時間可能就達到一天以上。

有些公司爲了避免這種狀況，會大手筆添購系統效能監控的工具，然實際上事情發生時不見得可以幫你追查到問題的癥結點，或者是該系統雖可細查系統狀況，但是卻會對系統本身造成極大的壓力。實際上大部分的公司需要的工具是「平時進行效能監控，遭遇問題時進行細部追蹤」的工具。

Foglight 是一個全方位的系統管理工具，可以同時掌控服務效能，亦可有效進行較細部的追查。何謂服務，試想，一線的系統監控人員並非系統專家，甚至不了解資料庫與 Application Server 系統，他可能只會知道客服人員所反映的是一個 XX 服務系統，此時對他而言最有幫助的訊息是，這個服務系統到底有包含哪些系統，例如：這個服務系統內，包含有一個 WebLogic Application Server，一個 Oracle 資料庫，而監控系統可以告訴他，到底是其中元件出了問題，而這個監控的功能有最好是圖形化的，可以一目瞭然。



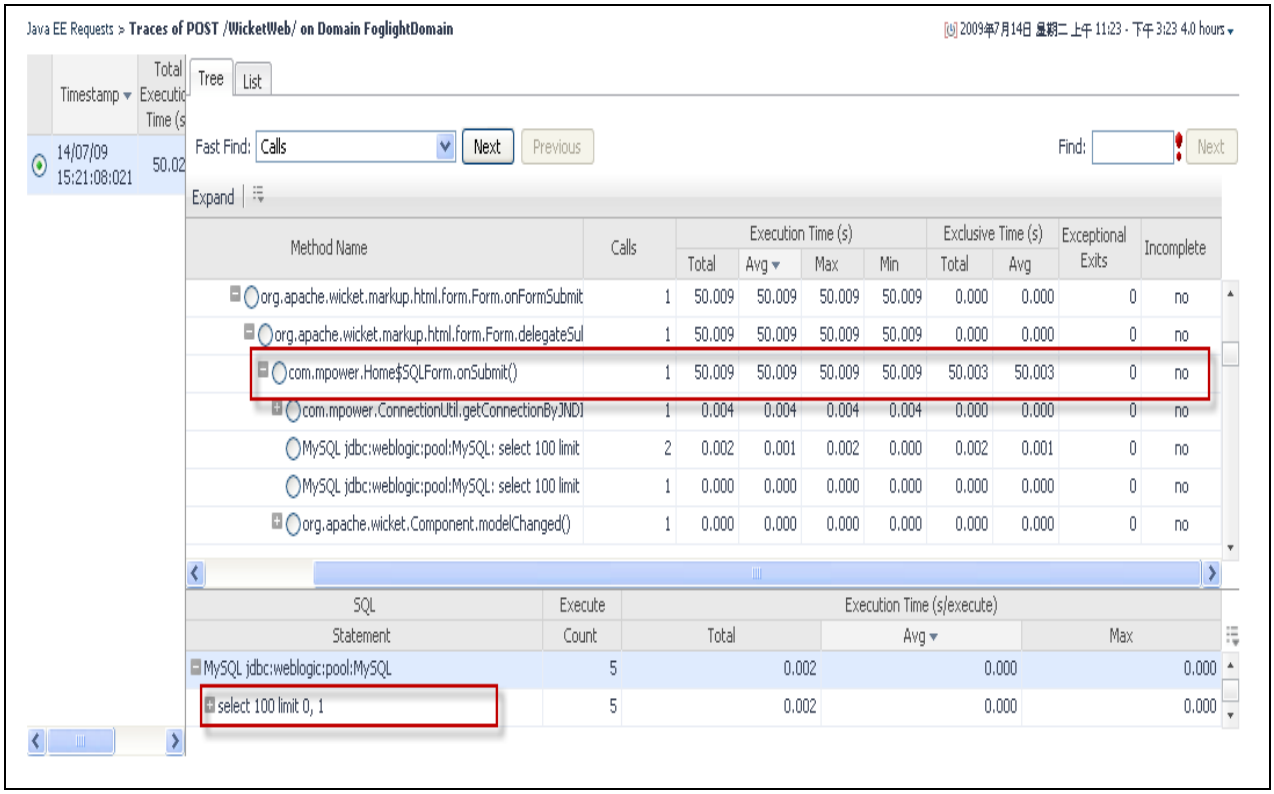


Foglight 提供的 Service 介面，可以輕易讓管理者將一個對外提供的服務的內容元件一一描繪在一個頁面上，並且以架構圖的方式展現出來，而一線系統監控人員由架構圖內子元件的狀態快速找到問題，一旦知道了問題之所在便可反映給該項元件的管理人員，由比較熟知該元件的人員來負責處理。這樣做的好處是可以快速定義異常所在。相較於以往，系統服務有問題時，所有

的子服務負責人均需排排站，每個負責人要做的事都是要證明自己負責的系統沒有問題，是使用消去法的方式來找到問題，兩者效益相差極大。

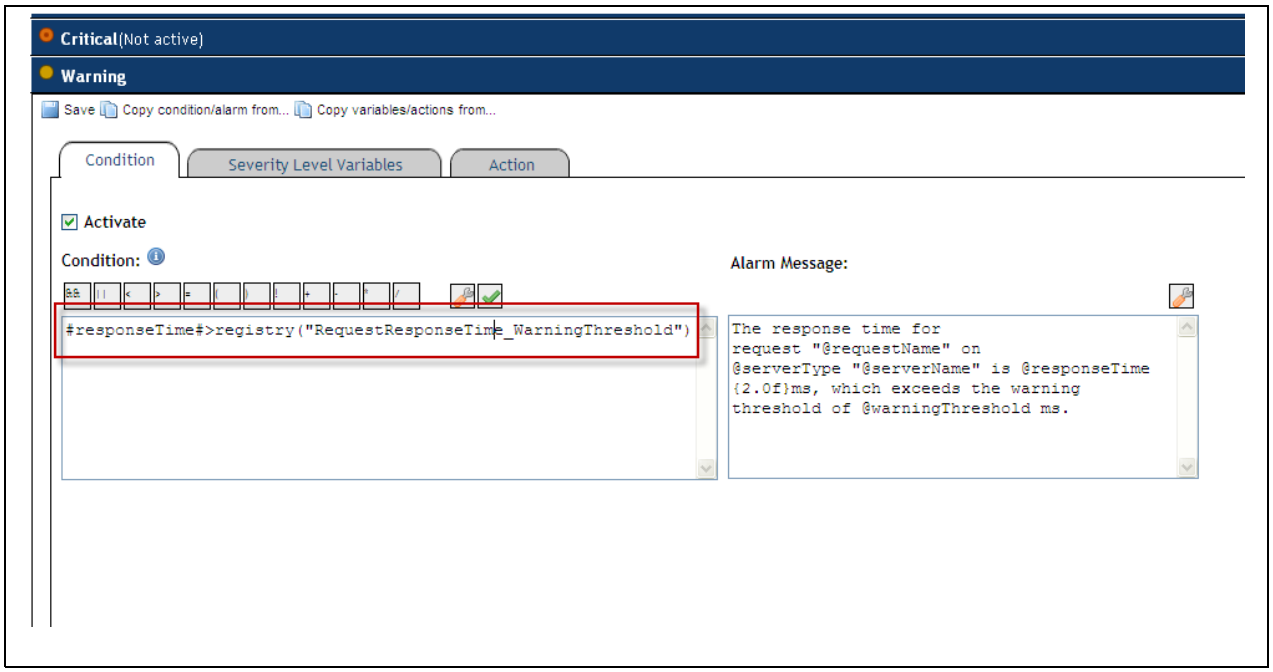
行文到此，我們由 Foglight 的協助，已經縮小了問題查找的範圍，再者，就是找到問題的點了。在此，筆者舉 Application Server 管理的例子來說明，有經驗的管理者常會發現，當使用者反應某個系統功能的速度變慢的時候，真正有問題的地方，常在於應用程式本身，而非是 Application Server 的問題，但是追查及舉證困難，常常要勞師動眾請 Server 管理者與程式開發人員開會、會後追蹤，又或者症狀只維持一小段時間，待要追查時已過了查找的黃金的時間點。

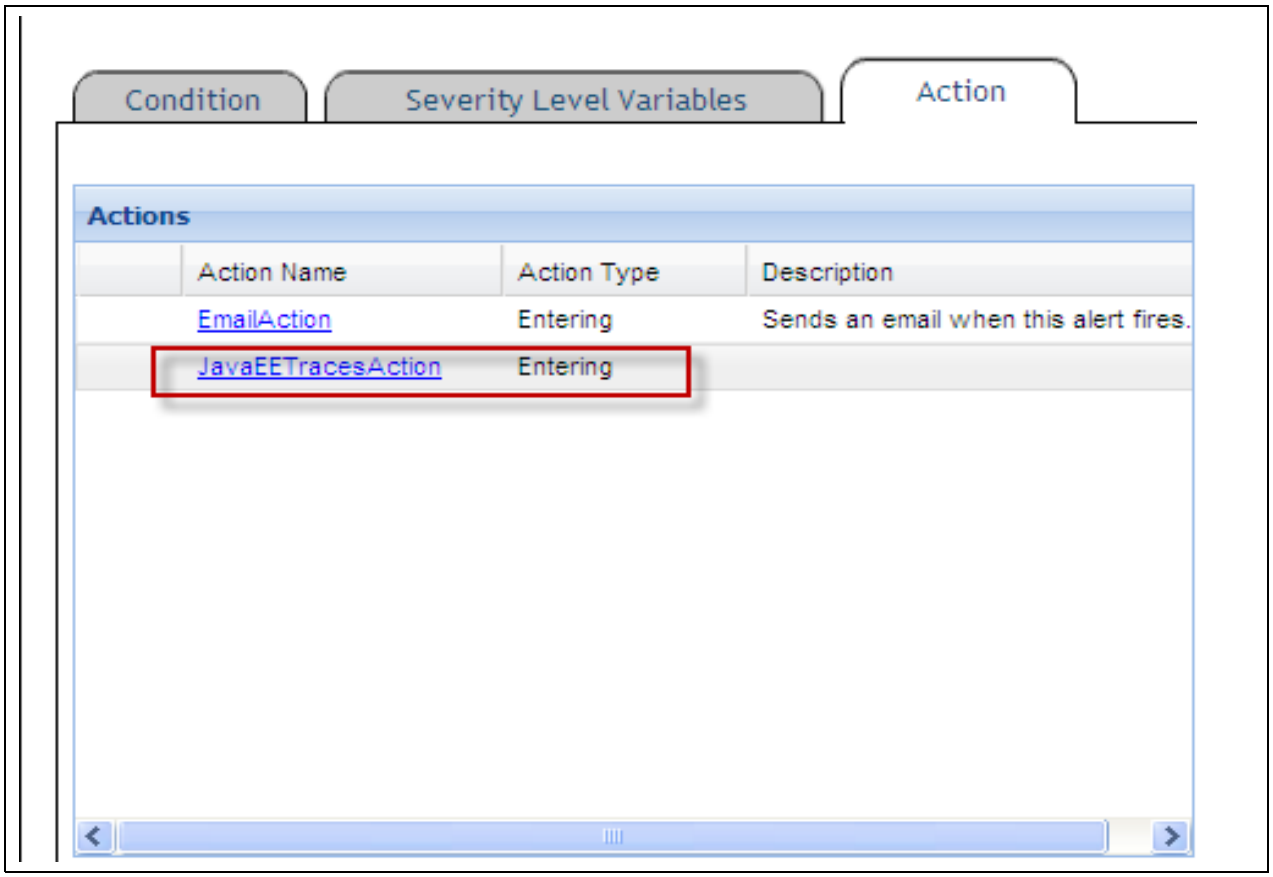
前文提到好的工具，遭遇問題時要可以進行細部追蹤，對 Foglight 而言，此功能甚至是自動化的，以 WebLogic Server 舉例來說，Foglight 發現到 WebLogic 在處理一個使用者的 Request 時所耗費的時間過長，就會自動啓動爲期 500 秒的追蹤，在該期間內所有 Request 中，並自其中保留 20 個取樣，追蹤該 Request 對應內部的應用程式呼叫的 Stack，如果有應用程式有執行 SQL 指令，也會一併取樣。



這個追蹤的功能對程式開發者也很有幫助，因爲其資訊詳細的程度可以到 Java Class 的方法，所以不需要在透過程式的修改如加上測試執行效能的 Flag，來進行痛苦除錯了。

而剛才所提到的自動化追蹤異常 Request 的原理其實就在 Foglight 內建的 Rule，每個預定義的 Rule 均有設定觸發的條件，符合條件則會發出警示並且執行預先定義的 Action，而 Request Trace 的功能就是眾多 Action 其中之一。由於 Request Trace 的行爲所消耗的系統資源較多，所以取樣數與追蹤的時間均有所限制，以免影響到系統的正常運作。





透過以上簡介的兩階段問題定位法，一定位，二追蹤，相信大多數的系統問題，可順勢排除，令系統的管理不再是件苦差事。