Glassfish 3.1.2.2 Web Service Memory Leak Workaround

|  |  |
| --- | --- |
| 問題說明 | AP02 server 運行一段時間後就會變慢或甚至無回應，觀察一陣子後發現memory有異常耗用的現象 (其它 servers大約不超過1GB，AP02卻可達到將近3GB)  cid:image004.png@01CF3FAF.95FD8400 |
| 診斷 | Dump heap memory:    jps -l: 列出目前執行中的 java process  grep –i asmain: 僅列出glassfish AP server process  jmap …: dump heap memory to file  Heap analysis tools:  JDK本身有提供 jvisualvm, jhat，推薦使用Eclipse Memory Analyzer Tool (MAT)  下載: <https://www.eclipse.org/mat/downloads.php> (選Stand-alone Eclipse RCP Applications)，已放到  \\192.168.204.10\TCCI\_File\軟體處\Public\Tools\Eclipse Memory Analyzer  解壓縮後，建議修改MemoryAnalyzer.ini，加上-Xmx3g  執行MemoryAnalyzer.exe，開啟ap02heap.hprof檔    可能要花好幾分鐘，建議在好一點的機器上做分析。    順便檢查是否有memory leak可疑份子    分析後顯示如上圖，total 1.6GB中ManagedBeanManagerImpl佔了1.2GB，非常可疑!  D:\temp\icons\fatcow\FatCow_Icons16x16\emotion_question.png shallow size/retained size  shallow size: instance本身佔用的memory size  retained size: 除了自己再加上被referenced到的object memory size  上圖shallow size 56B，retained size 1.2G，持續追查到底是哪些object佔了那麼大的memory。    點選Dominator Tree，顯示如下:    展開ManagedBeanManagerImpl，看看裡面有什麼吧!    一堆OrgUserService(web service)，似乎找到元凶了，但為什麼呢?  看來只好請Google大神了，試試找 glassfish memory leak看看    第2筆滿符合我們的case，點進去看看內文:  主要內容是做了一個dummy web service並且用JMeter去做壓力測試，得到的結果跟我們情況相同。  在底下的討論串中有人提到在glassfish 3.1.2時就有這個bug了，但要在glassfish 4才會修改。看來root cause找到了! |
| 解決方式 | 在上述bug report中有提到兩種workaround解決方式   1. 刪掉beans.xml (不使用CDI功能)。   但我們的foundation jar已經有使用CDI了，無法使用這方法。   1. 使用@Stateless並加上glassfish-ejb-jar.xml去設定web service設定(原本只用annotation即可)   修改如下:    增加@Stateless宣告，變成了EJB了  加上glassfish-ejb-jar.xml檔案 (沒有此檔案時，呼叫service會回傳404 error)    重新deploy至AP02，目前沒有再發生memory異常耗用的現象了。 |