

品質控制流程應用於雲端服務監控機制之研究

陳奕明 楊采薇
國立中央大學資訊管理學所
cym@cc.ncu.edu.tw
osmund1135@hotmail.com

摘要

雲端應用服務的品質會受到底層 IaaS 所提供的服務品質影響，當服務發生異常時，是否能即時辨別問題的來源，會影響問題解決的速度及應用服務的品質。目前現有的雲端監控功能，若運用到雲端應用服務，僅能獲得服務或資源狀況，無法辨別服務問題來源。故本研究以 CSEP 雲端安全實驗平台為例，轉換傳統品質控制流程步驟，將之運用於雲端服務監控機制，並以定期監控發現服務問題、追蹤監控辨別問題來源。本機制經實驗證明可有效偵測雲端應用服務 CSEP 共六種主要服務之服務問題及雲端基礎設施供應商 SAMEVED API 服務異常情形，並能即時辨別錯誤來源，幫助提升問題發現與解決速度，提供未來雲端應用服務進行監控設計時作為參考。

關鍵詞：雲端服務、雲端環境監控、品質控制、安全實驗平台

Abstract

Cloud application services have become more popular and will be affected by IaaS's quality of service. The problems cloud application services encountered be recognized or not will affect problem-solving time and quality of service. There are some researches and tools for cloud monitoring, and using these methods for cloud application services can only check the status of the service but not recognize the problems. We proposed a mechanism that used traditional quality control process for cloud services monitoring, and used CSEP Cloud Security Experimental Platform as example. Our mechanism including a period monitor and a trace monitor for finding problems. Our mechanism can recognize six CSEP main service problems and IaaS SAMEVED's API service problems, and provides some error messages to CSEP admin and users for problem solving and giving another cloud application services monitoring as a reference.

Keywords: Cloud Services, Cloud Monitoring, Cloud Security Experimental Platform, Quality Control

1. 緒論

近年來雲端運算 (Cloud computing) 興起，雲端應用服務隨之成長。雲端應用服務易受到底層 IaaS 服務影響[1]，當發生服務異常時能否即時辨別問題來源，會影響問題解決速度及服務品質，因此雲端應用服務品質監控是一個很重要的議題。目前現有的雲端監控，主要以 IaaS 資源、安裝於其上之應用程式及 SLA 監控為主。若將現有的 IaaS 監控方式運用到雲端應用服務，僅能獲得服務或資源狀況，無法辨別服務問題來源。故為了確保雲端應用服務的品質，本研究以 CSEP 雲端安全實驗平台為例，所設計的監控機制主要分成兩個部分：追蹤監控使用者使用 CSEP 雲端服務的過程，對於服務提供的狀況進行監控；對 IaaS 服務進行定期監控，以掌握底層服務狀況。本研究目的包含以下兩點：(1) 運用傳統品質理論，對 CSEP 雲端應用服務進行分析，並將品質控制概念運用於雲端服務監控機制，以達到服務問題之辨別。(2) 以雲端安全實驗平台為例，設計一套監控機制，即時發現與辨別雲端安全實驗平台服務異常問題來源，提高系統穩定度以達成服務品質控制。

2. 相關研究

本節分別介紹品質管理中品質管制的相關理論及雲端服務監控中相關研究及作法。本研究有別於針對 IaaS 基礎資源使用率及應用程式的監控方式，採用監控雲端應用服務情況，進而監控到應用服務會使用到的 IaaS 服務，並運用朱蘭品質三部曲中的品質控制流程，轉化成適用於雲端應用服務的品質控制流程用於 CSEP 系統分析，並作為本研究監控機制與錯誤辨別設計的基礎。

2.1 品質管理與控制

品質控制是用於確保在產品製造或服務提供時，能滿足所定義的品質標準或達成顧客的需求的一套程序[2]。為了達到顧客對於品質的要求，透過監視品質形成過程，進而消除過程中各階段所引起不符合品質標準或使顧客不滿的因素[3]。傳統的品質控制通常涵蓋於品質管理理論中，而品質管理是為了保障產品或服務的品質達到訂定的品質標

準，所進行的各種管理活動[4]。

在傳統品質管理有不少著名的理論，PDCA 管理循環[5]透過計畫、執行、查核及處置四步驟的循環，用於確保目標達成，促使品質持續改善，經常結合統計工具用於解決品質管理問題。費根堡全面品質管理[6]對於品質管制工作給予四步驟的說明，主要為分析實際狀況與管制目標之差距，若有不符目標的狀況便予以修正並重新思考目標的訂定基礎。朱蘭品質三部曲[7]品質三部曲為一管理循環包含品質規劃、品質控制及品質改善，其中品質控制注重於選擇控制單位並進行實際狀況測量，當測量不符合所訂定之品質目標時採取管制行動。六標準差管理**錯誤！找不到參照來源。**透過界定、測量、分析、改善及控制五步驟，主要用於改善現有的流程，當確定欲改進的目標並了解現有質量水平後，利用統計學工具分析影響品質的因素，執行改善方案並對於新的流程進行監控使其發揮功效。品質三環[8]在品質三環中包含品質管制、品質保證及品質管理，其中品質管制活動主要是透過抽樣製作管制圖的方式來減少製程上的變異。

PDCA 管理循環、朱蘭品質三部曲之品質控制、費根堡全面品質管理之品質管制與六標準差管理，其品質控制的流程大同小異，首先都從確定目標開始、進行實際測量、比對測量結果與目標、對比較後的差異進行改善或其他處理。而品質三環之品質控制則從管制圖的角度對品質進行控制與管理。由於管制圖的概念是希望減少製程上的變異，其目標並不符合本研究監控機制欲達成主動發現服務問題的目標，因此比較其餘理論後採用步驟說明較為詳細之朱蘭品質三部曲中的品質控制流程，進行轉換並運用於本研究雲端服務監控機制的設計。

2.2 雲端服務監控

為了確保雲端服務的品質，雲端服務供應商會與其顧客簽訂服務層級協議 (Service Level Agreement, SLA)。SLA 是由服務的使用者與提供者所共同訂定的協議，協議內容包含服務品質、權利義務歸屬、服務目標及雙方的預期和責任定義。常見的系統服務關鍵績效指標如系統可用性、系統回復性、系統回應時間、網路 QoS (Quality of Service) 等。

目前有不少研究針對 IaaS 供應商是否有滿足 SLA 協議內容而進行雲端基礎設施服務的環境資源監控。High-level market-oriented Cloud architecture[10]從市場導向的架構來看資料中心或雲端環境之 SLA 資源配置，判斷目前雲端的資源是否可達成使用者的需求，以及使用者使用了多少雲端資源並且如何計價。FOSII[11]使用 IBM 提出的 MAPE-K 自我管理流程[12]運用於其監控機制，利用 DeSVi[13]架構檢查是否違反 SLA，並將監控結果反映回應用程式及雲端基礎設施資源上以達成

一個自我管理的循環。Partial ASP control architecture[14]從雲端應用服務提供者的角度來監控雲端基礎設施資源，相較架設於 IaaS 環境之監控機制能更實際的評估雲端基礎設施服務供應商是否滿足 SLA。

[10][11][14]皆從監控雲端服務層級協議的角度出發，但他們欲達成之目標並不相同。[10]注重在監控時能即時辨別 SLA 的違反情形，並可實踐雲端資源的自我管理且將資源依據監控分析後的結果自動進行調整。[11]則鎖定在使用者下達需求時，依據資源監控的結果評估現有的使用者其資源供給是否皆符合 SLA，若資源仍能供應新進使用者的需求，才決定接受新來的需求並執行。[14]從使用 IaaS 之雲端應用服務的角度進行雲端基礎設施資源的監控，以更準確的評估雲端基礎設施服務供應商是否滿足 SLA。本研究對於雲端基礎設施服務之監控方式類似於 Partial ASP control architecture，從雲端應用服務的角度出發更準確地監控資源給予的狀況。

3. 雲端應用服務監控機制

在提出本研究之雲端應用服務監控機制前，首先須了解 CSEP 平台的系統設計及運作方式，因此在 3.1、3.2 節介紹 CSEP 的系統架構與分析，3.3~3.5 節針對問題並提出本研究之監控架構與流程，包含 CSEP 服務追蹤監控及 API 服務定期監控機制。

3.1 CSEP 系統架構

CSEP 使用 Django Web 架構[15]及底層雲端基礎設施服務提供者 SAMEVED[16]提供 Xen[17]之虛擬環境進行實作。CSEP 平台所提供之安全實驗案例會先製作好並以映像檔 (image) 方式儲存於 SAMEVED，這時的映像檔稱為樣版映像檔 (template)。當使用者進入 CSEP 平台並選擇想進行的實驗時，網頁介面會透過呼叫 SAMEVED API 的方式先複製一份樣版映像檔，當使用者開機後便可透過 Windows 遠端桌面協定 (RDP) 或 SSH 連線至虛擬機器開始進行實驗，詳細流程如圖 1。

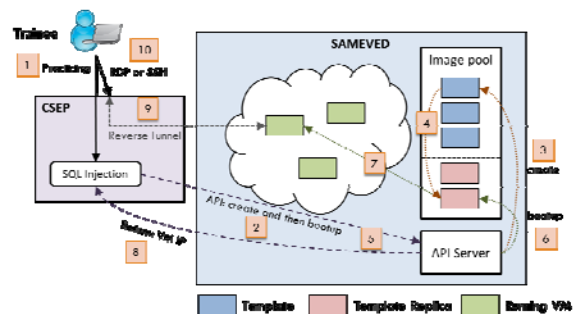


圖 1 CSEP 平台流程

為了節省資源，當 CSEP 平台偵測到使用者已一段時間未有操作虛擬機器的行為時，CSEP 平台會自動下達終止的 API 指令，當使用者打算繼續進行實驗，CSEP 平台會再下達恢復的 API，讓虛擬機器恢復運作並復原到機器終止前的狀況。最後當使用者完成案例時，系統會自動將案例刪除並於資料庫紀錄實驗已完成。

3.2 CSEP 系統分析

本節使用朱蘭品質三部曲之各流程，針對品質規劃、控制及改善分析 CSEP 平台在品質管理的過程中需依序進行的步驟。在品質規劃的部份，CSEP 平台的目標為透過雲端服務的方式，提供資訊安全受訓人員可實際進行安全實驗操作的環境。欲達到上述目標，CSEP 平台須具備以下之服務項目：(1) 建立安全實驗，需使用到 SAMEVED Clone API，(2) 顯示實驗資訊，含實驗機器列表、實驗手冊、配置、原理、步驟，(3) 操作實驗機器，需使用到 SAMEVED Boot up, Shutdown API，(4) 恢復被終止實驗，需使用到 SAMEVED Resume API，(5) 刪除實驗，需使用到 SAMEVED Delete API。

在品質控制過程中，會透過監視品質形成的過程中是否有產生不符合品質標準的因素，並將其消除以達成品質目標。以 CSEP 平台為例，CSEP 平台服務品質形成的過程即包含上述五個流程的運作，因此對於每個流程皆應透過品質控制流程，對服務產生的過程進行監控。在品質控制的過程中，每次會選定一個控制對象，並設定在控制過程中要檢查的項目。以第一點建立安全實驗為例，建立安全實驗為控制對象，而在建立安全實驗中尚包含取得實驗案例資訊、使用 Clone API 建立實驗機器、及儲存實驗機器相關資訊三個步驟，因此這三個步驟便作為在控制建立安全實驗的過程中的檢查項目，而檢查步驟便為這三個步驟的執行順序。透過本研究追蹤監控機制進行實際的品質狀況檢查，並透過問題辨別元件分析目標與實際狀況的差距，詳細監控設計於3.4節進行說明。最後的管制行動則包含使用者會於 CSEP 錯誤頁面上看到提示訊息、CSEP 管理者可於監控頁面查看問題狀況、CSEP 管理者人工檢查問題，確認問題實際狀況以進行後續處理。

在品質改善的部分，首先透過監控資料評估經常發生問題之項目，以確認改善項目。透過追蹤監控可發現問題來源，經由人工檢查確認後找出解決方式。解決問題且不引起其他問題，以證明解決方式有效。最後透過持續進行監控，確認問題是否已改善並維持改善成果。

3.3 監控架構

為了達成能主動發現服務問題並得知問題來源，本研究監控機制之監控目標為以下兩點：(1) 以追蹤使用者網頁操作的方式，當 CSEP 平台發生

服務問題時能即時得知問題狀況，並取得問題來源。(2) 以定期監控的方式，當 SAMEVED API 發生服務問題時能即時發現。

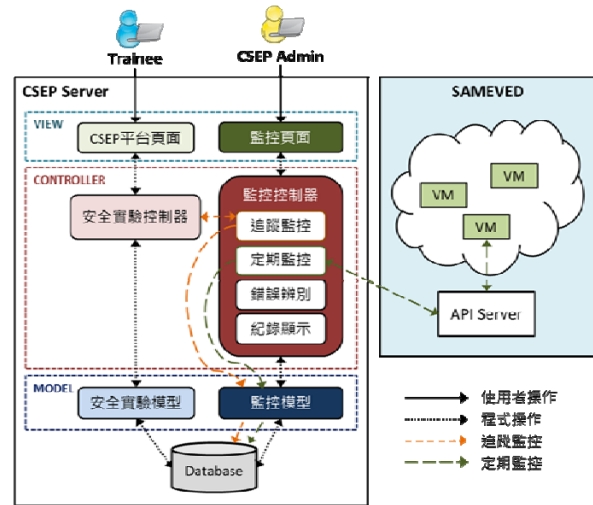


圖 2 本研究雲端應用服務監控架構

本研究監控機制從雲端應用服務 CSEP 平台的角度出發進行監控，由於 CSEP 使用 Django 架構進行實作屬於 Model-View-Controller (MVC) 架構，其架構如圖 2。在 CSEP Server 中可以分成 CSEP 平台與服務監控平台兩部分，以下將針對各元件進行說明：

- CSEP 頁面：為 CSEP 平台提供給使用者進行實驗操作之網頁介面，由 HTML 檔案組成。
- 安全實驗控制器：當使用者進行安全實驗時對 CSEP 頁面進行操作後，會將使用者需求交由安全實驗控制器進行處理，例如建立實驗、顯示實驗詳細資訊、開啟或關閉實驗虛擬機器等。
- 安全實驗模型：當需要儲存或取得與安全實驗相關資料時，會透過安全實驗模型對資料庫進行存取。
- 監控頁面：提供給 CSEP 管理者對於 CSEP 平台之服務問題或 SAMEVED API 服務問題一個即時顯示的頁面，由 HTML 檔案組成。
- 監控控制器：為本研究核心，包含以下四種控制元件。(1) 追蹤監控：對 CSEP 平台使用者操作進行追蹤，將於3.4節詳述。(2) 定期監控：進行定期監控作業，如對 SAMEVED API、虛擬機器資源使用狀況進行定期監控，將於3.5節詳述。(3) 錯誤辨別：進行 CSEP 平台或 SAMEVED 服務問題異常時的辨別。(4) 紀錄顯示：當管理者想查看監控資訊或偵測到服務發生問題時，該元件會呼叫監控模型取得相關資訊，並回傳給監控頁面並顯示。
- 監控模型：當需要儲存或取得與服務監控相關資料時，會透過監控模型對資料庫進行存取。

3.4 追蹤監控

追蹤監控會記錄功能處理的時間、回傳值及錯誤訊息，因此在每項功能中捕捉例外狀況之處會設計追蹤點。當 CSEP 平台使用者的操作觸發到追蹤點時，便會記錄錯誤訊息並回傳給監控控制器中的追蹤監控元件進行處理。以建立實驗為例，以下為虛擬程式碼：

```

1 createExperiment()
2     startTime = time.now()
3     Start of create experiment
4     Exception Case DoesNotExist
5         msg = experiment create error explanation
6         saveException(startTime, time.now(), 1, msg)
7     End of create experiment
8     saveLog(startTime, time.now(), 0, "Experiment
    created successful")
9     return to experiment detail page
    
```

當使用者對 CSEP 平台進行操作，程式執行進入建立實驗功能時會先記錄當下時間(第 1~2 行)，在建立實驗的過程中(第 3~7 行)若有觸發到例外狀況，如找不到使用者所選擇的案例內容，此時會觸發追蹤點並傳遞功能開始執行的時間、當下時間、回傳值及錯誤訊息給監控控制器中的追蹤監控元件(第 4~6 行)。若建立實驗功能成功執行完畢，則會於功能結束前將資料傳遞給監控控制器進行儲存(第 9 行)。以下為監控控制器中追蹤監控之虛擬程式碼：

```

1 saveLog(startTime, endTime, returnCode, msg)
2     duration = calculateDuration(startTime,
3     endTime)
4     save returnCode, msg, duration to database
5
6 calculateDuration(startTime, endTime)
7     duration = endTime - startTime
    return duration
    
```

當資料由安全實驗控制器傳遞至監控控制器後，會進行計算處理時間的動作(第 2、5~7 行)，並將回傳值、成功或錯誤訊息及計算後的處理時間儲存作為一筆監控資料(第 3 行)。當追蹤監控機制要辨別服務問題及來源時，會先從回傳值進行判斷，如果回傳值為 0 代表成功、1 為失敗代表有服務異常情況發生。此時追蹤監控機制會以錯誤來源對照表去比對錯誤訊息，並以此取得問題來源，錯誤來源對照表如圖 3 所示。

項目	錯誤訊息	來源		
Name	Message	App	File	Function
恢復實驗失敗	Resuming	ade	views.py	detail
顯示實驗頁面失敗	show detail	ade	views.py	detail
建立實驗失敗	clone	ade	views.py	expCreate
刪除實驗失敗	delete	ade	views.py	expDel
Reverse tunnel建立失敗	tunnel	ade	vagent_server.py	ctport
網頁顯示 IP、Port 失敗	broadcast	ade	vagent_server.py	ctport
未獲存取實驗頁面	access	ade	views.py	detail
實驗頁面不存在	DoesNotExist	ade	views.py	detail
關機動作處理失敗	shutdown	ade	views.py	vmact
開機動作處理失敗	bootup	ade	views.py	vmact

圖 3 CSEP 錯誤來源對照表

3.5 定期監控

在進行底層 API 定期監控時，會以一個永遠成立的 while 迴圈搭配 sleep function 達成定期呼叫機制。當進入迴圈後，先設定三種 API 呼叫模式值為 False。第一類型 API 呼叫模式為操作實驗時最基本的 API，第二、三類型則以 CSEP 平台的附加功能檢查為主，如圖 4。

SAMEVED API		功能說明
第一類型	Clone	備載安全實驗環境之虛擬映像檔(template)，使其成為可供使用者操作之實驗虛擬機器
	Boot up	將實驗虛擬機器開機
	Shutdown	將實驗虛擬機器關機
第二類型	Delete	將實驗虛擬機器映像檔刪除
	Suspend	將實驗虛擬機器進行停止
	Resume	將實驗虛擬機器恢復操作並恢復停止前的虛擬環境
第三類型	GetIP	取得實驗虛擬機器 IP
	Monitor Start	開始監控實驗虛擬機器之 CPU、記憶體、網路、及硬碟使用狀況
	Monitor List	因每台實驗虛擬機器之配置皆不相同，故API可取得受監控的項目列表
	Monitor Result	取得實驗受監控項目所感測到的數值

圖 4 SAMEVED API 列表

由於不同類型之 API 使用頻率差異甚大，因此將 API 進行分類。每個類型中只要其中一個 API 最後呼叫時間與當前時間差超過所設定的 threshold，則設定該類型 API 呼叫模式值為 True。接著判斷如果三種 API 呼叫模式中有一種被設定為要檢查(值為 True)，則會啟動 API 呼叫機制。完整呼叫流程如圖 5，會從編號 1~9 依序呼叫；若僅啟動第二類型檢查，呼叫順序為編號 1~2、6~9；若僅啟動第三類型檢查，呼叫順序為編號 1~5、8~9。

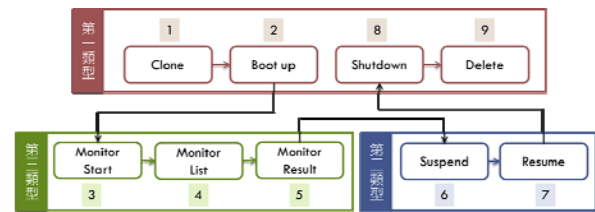


圖 5 SAMEVED API 呼叫流程

4. 實驗結果與討論

在本節主要對於 CSEP 及 SAMEVED API 服務狀況進行異常察覺與來源區別，以及對 SAMEVED API 定期服務監控機制進行實驗。CSEP 服務異常狀況察覺與區別問題來源之實驗項目，以達成 CSEP 平台服務目標所包含之服務為主，包含建立實驗失敗、開機動作處理失敗、關機動作處理失敗、網頁顯示 IP、Port 失敗、恢復實驗失敗、刪除實驗失敗。

4.1 實驗配置與方法

本研究監控機制建置於 CSEP 平台上，CSEP 使用 AMD Athlon(tm) 64 X2 Dual Core Processor

3800+及 RAM 3G，作業系統為 Ubuntu 10.04。雲端基礎設施環境為 SAMEVED，使用 Xen 作為開發環境。本研究監控機制設計為能主動發現服務異常問題，實驗主要分成兩部分如下：(1) 服務異常察覺與來源區別：對於是否能即時察覺服務異常狀態並分析來源進行實驗，(2) 定期服務監控：對於 API 定期服務監控機制是否能發現服務問題進行實驗。

4.2 實驗結果與討論

以建立實驗失敗為例，進入 CSEP 平台首頁，當使用者點選 CSEP 平台可進行實驗列表中的實驗項目後 (圖 6a)，CSEP 平台會開始進行建立實驗的動作。當實驗建立成功時會顯示實驗詳細資訊如實驗機器列表、實驗原理、配置、手冊、操作步驟等；當實驗建立失敗時，使用者會看到已通報管理員請稍後再試之提示訊息 (圖 6b)。



圖 6 建立實驗失敗使用者畫面



圖 7 建立實驗失敗監控平台畫面

此時監控平台可以看到該錯誤的相關資訊如錯誤發生時間、來源、內容，在此實驗中實驗建立失敗的原因為給予 SAMEVED API 的參數並非正確的實驗樣版映像檔編號，因此 API 無法將實驗所需之虛擬機器進行複製的動作，因而造成建立實驗失敗。此時在監控平台頁面可同時看到 CSEP 及 SAMEVED 服務異常的項目說明 (圖 7)。

在定期服務監控的部分，主要針對十個 API 項目進行定期監控，設定每隔一段時間就會進行 API 呼叫，並記錄呼叫時間及回傳訊息，圖 8 為 CSEP 系統中的 API 定期呼叫紀錄，其詳細資訊可於 CSEP 的管理介面查看。每次啟動定期呼叫機制時，其特色為在短暫的一至兩分鐘以內會有連續呼叫一系列 API 的動作。

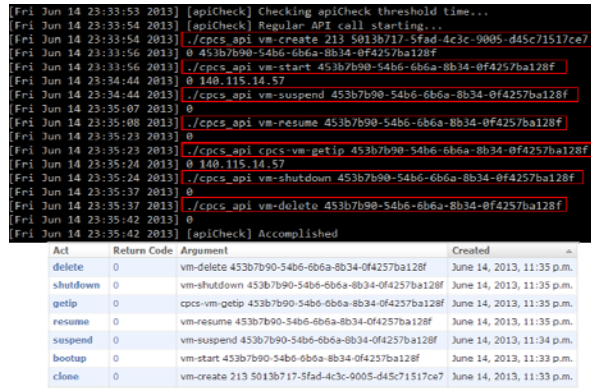


圖 8 API 定期呼叫對照

每次進行定期監控時會留下紀錄，可於監控平台查看，對於每次監控皆會顯示處理時間 (秒) 以及成功或失敗。圖 9 為 API 的定期監控紀錄，可看到編號 15 於 June 16, 2013, 7:09 p.m. 時定期監控機制運行失敗，對於運行失敗的條列可以查看詳細的錯誤資訊。

編號	監控時間	成功/失敗	處理時間	Detail
13	June 16, 2013, 12:48 p.m.	成功	4.244265	
14	June 16, 2013, 12:42 p.m.	成功	3.424214	
15	June 16, 2013, 7:09 p.m.	失敗	28.737796	
16	June 16, 2013, 7:20 p.m.	成功	3.612225	
17	June 16, 2013, 8:15 p.m.	成功	4.836303	

圖 9 API 定期呼叫紀錄

Time	Action	Arguments	Return code	Return Message
June 16, 2013, 7:10 p.m.	bootup	vm-start 1a14f5c-b4c-fd21-ab95-34ba652be7d0	1	[1 Error: Could not connect to XenServer Controller.v]
June 16, 2013, 7:10 a.m.	suspend	vm-suspend 1a14f5c-b4c-fd21-ab95-34ba652be7d0	1	[1 Could not connect to XenServer Controller.v]
June 16, 2013, 7:10 p.m.	resume	vm-resume 1a14f5c-b4c-fd21-ab95-34ba652be7d0	1	[1 Could not connect to XenServer Controller.v]
June 16, 2013, 7:10 a.m.	shutdown	vm-shutdown 1a14f5c-b4c-fd21-ab95-34ba652be7d0	1	[1 Could not connect to XenServer Controller.v]
June 16, 2013, 7:10 p.m.	delete	vm-delete 1a14f5c-b4c-fd21-ab95-34ba652be7d0	1	[1 Could not connect to XenServer Controller.v]

圖 10 API 定期呼叫錯誤紀錄詳細資訊

詳細的錯誤資訊如圖 10 所示，可以看到圖 9 編號 15 的錯誤運行紀錄中為類型一的 API 呼叫模式，而這五筆 API 呼叫錯誤情形皆為 SAMEVED API Server 回傳 Could not connect to XenServer，此錯誤情形會發生於 SAMEVED API Server 系統異常時，因此本研究 SAMEVED API 定期監控機制確實可發現服務異常問題。

4.3 小結

根據以上兩節的實驗結果，證明本研究雲端服務監控機制可有效偵測雲端應用服務 CSEP 平台共計六種主要服務之服務問題及雲端基礎設施 SAMEVED API 服務異常情形。當服務發生問題時能提供錯誤訊息提示給使用者，並能即時辨別錯誤來源給服務管理者。在定期服務監控機制方面，能針對雲端服務進行定期監控，主動發現服務異常問

題並提供給管理者進行處理。

由於本監控機制對於使用者行為進行記錄，因此可以使用者操作為基準，計算出 CSEP 平台之可用性，並且於未來研究可以提高 CSEP 平台可用行為基礎進行品質改善的設計。圖 11 為從 June 15, 2013 至 July 24, 2013，統計 CSEP 平台五項主要操作成功或失敗次數，期間內共 382 項紀錄，其中 316 項為操作成功、66 項為失敗，其可用性為 83%。從圖 38 可看出當 CSEP 服務項目牽涉到 IaaS API 的服務時，如建立安全實驗、操作實驗機器、恢復被終止實驗、及刪除實驗，其發生錯誤的機率是較高的，因此雲端應用服務之監控是有其必要性的。

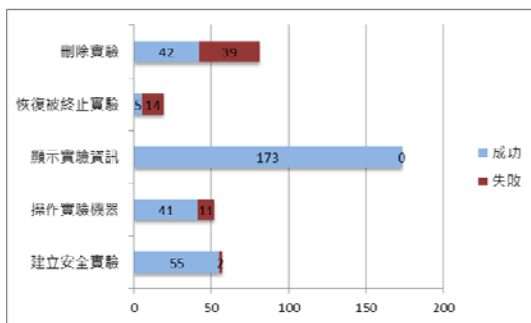


圖 11 June 15, 2013~July 24, 2013 監控記錄統計

5. 結論與未來研究方向

5.1 研究結論與貢獻

隨著雲端運算的興起，雲端應用服務也跟著成長。雲端應用服務使用 IaaS 作為底層的服務提供，並於上方建構應用服務。目前已有的雲端監控，主要以 IaaS 資源、安裝於其上之應用程式及 SLA 監控為主，而非針對雲端應用服務進行監控，無法滿足雲端應用服務在異常時的偵測及問題辨別。因此本研究結合傳統的品質控制流程，設計一套雲端服務監控機制，並以定期監控及追蹤監控兩種方式發現服務問題，經實驗證明能有效發現服務異常狀況，幫助管理者掌握服務問題，進而提供使用者更穩定的服務品質。綜合上述，本研究主要貢獻有以下兩點：(1) 轉換朱蘭品質三部曲中品質控制流程，運用於雲端應用服務品質控制，以達到服務問題之辨別。(2) 運用品質控制概念並針對 CSEP 雲端應用服務的特性，提出一套監控機制讓服務提供者能即時得知服務狀況。

5.2 未來研究方向

未來可研究方向主要為兩部分：(1) 雲端服務全面監控及 (2) 雲端服務品質改善。在雲端服務全面監控的部分，雲端應用服務位於中間的角色，對外提供使用者服務，內部則受到 IaaS 服務狀況的影響，若能掌握 IaaS、雲端應用服務、以及使用者

的狀況，便能達到更全面的監控，盡可能地減少服務運行的不確定因素。對於雲端服務品質改善，當雲端服務發生異常的狀況時，能依據監控平台顯示之問題的嚴重程度、發生頻率、及問題來源，進行改善項目的訂定並找出有效的解決方法，提高 CSEP 平台可用性以達成雲端服務品質的改善。

參考文獻

- [1] DigiTimes 電子報，邁向混合雲 展現雲端效益，2013 年 4 月。2013 年 7 月 10 日，取自：http://www.digitimes.com.tw/tw/b2b/Seminar/shwnws_new.asp?CnID=18&cat=99&product_id=051A20326&id=0000331141_FNO017TZ5U_T7E30PTWRJY
- [2] WhatIs.com, Definition Quality Control (QC). Retrieved July 8, 2013, from: <http://whatis.techtarget.com/definition/quality-control-QC>
- [3] MBA 智庫百科，質量控制(quality control)。2013 年 7 月 8 日，取自：http://wiki.mbalib.com/zh-tw/%E8%B4%A8%E9%87%8F%E6%8E%A7%E5%88%B6#_note-0
- [4] Quality Management, Investopedia. Retrieved July 8, 2013, from: <http://www.investopedia.com/terms/q/quality-management.asp>
- [5] 全面品質管理 Total Quality Management Chap2 品質大師的品質理念。2013 年 7 月 8 日，取自：<http://www.cyut.edu.tw/~lschen/tqm/TQM-ch2.pdf>
- [6] 楊素芬，品質管理 二版，華泰文化。
- [7] J.M. Juran, "The Quality Trilogy A Universal Approach to Managing for Quality," ASQC 40th Annual Quality Congress, 1986.
- [8] 潘浙楠，華泰文化，品質管理：理論與實務（二版）chapter 16 六標準差（Six Sigma）的策略架構與實施模式。
- [9] S. Thomas Foster，智勝文化事業有限公司，品質管理（再版）。
- [10] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg and Ivona Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems, Volume 25, Issue 6, pp. 599-616, June 2009.
- [11] Vincent C. Emeakaroha, Marco A. S. Netto, Rodrigo N. Calheiros, Ivona Brandic, Rajkumar Buyya, César A. F. De Rose, "Towards autonomic detection of SLA violations in Cloud infrastructures," Future Generation Computer Systems, Vol. 28, No. 7., pp. 1017-1029, July 2012.
- [12] IBM, "An Architectural Blueprint for Autonomic Computing," Autonomic Computing White Paper, June 2006.
- [13] V.C. Emeakaroha, R.N. Calheiros, M.A.S. Netto, I. Brandic, C.A.F. De Rose, "DeSVi: an architecture for detecting SLA violations in cloud computing infrastructures," in: Proceedings of the 2nd International ICST Conference on Cloud Computing, CloudComp'10, 2010.
- [14] Emiliano Casalicchio, and Luca Silvestri. "Autonomic Management of Cloud-Based Systems: The Service Provider Perspective," Computer and Information Sciences III, pp 39-47, 2013.
- [15] Django Software Foundation. Retrieved July 8, 2013, from: <https://www.djangoproject.com/>
- [16] SAMEVED. [Online] Available: <http://www.openfoundry.org/of/projects/1657>
- [17] Xen. [Online] Available: <http://xen.org>