

基於微服務架構的智慧管理系統之設計

Design of Intelligent Management System Based on Microservice Architecture

報告者: 張俊彥

指導教授: 張保榮 教授

OUTLINE

1. Introduction
2. Research Method
3. Experimental Results
4. Conclusion

1. Introduction

3

背景

- 近年來企業數位轉型已成為必須面對的課題，尤其在2020年COVID-19疫情爆發後，疫情期間隨著防疫政策、居家辦公、遠端視訊會議等措施，大家生活型態及工作型態的轉變所帶來的衝擊，更加速了各國產業推動數位轉型的腳步。
- 雲原生平台是數位專案的重要核心，微服務架構不但可以解決傳統實體機器設定管理的運維問題，經由統一自動化的部署建置、升級、管理機制，高可用度的設計大幅降低硬體設備及系統運維的成本，也有助於提升應用軟體服務的開發速度。

4

雲原生

- 雲原生運算基金會(CNCF)於2018的定義:「雲原生技術有利於各組織在公有雲、私有雲和混合雲等新型動態環境中,構建和運行可彈性拓展的應用」。
- 發展至今,對於雲原生定義的說法很多,只要是運用雲原生架構的各種技術組合所發展出來的解決方案,例如:容器、CI/CD、微服務、聲明式API..等,都可以說是雲原生應用,是一個新形態現代系統開發的概念。
- 雲原生與數位轉型關係密不可分,因雲原生技術本身屬於架構的改變,並非所有舊系統都能轉換在其架構上運行,轉換過程必須考量風險,循序漸進的以階段性的方式完成轉型。

5

雲原生(cont.)

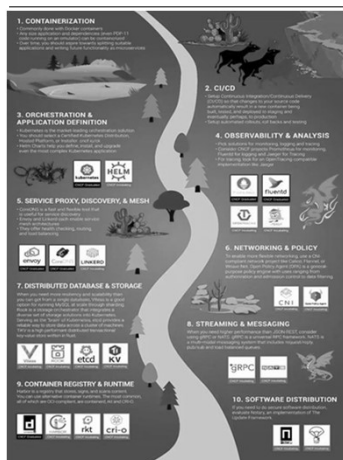
source : <https://www.cncf.io/>

Fig. 1. CNCf Cloud Native Trail Map 1

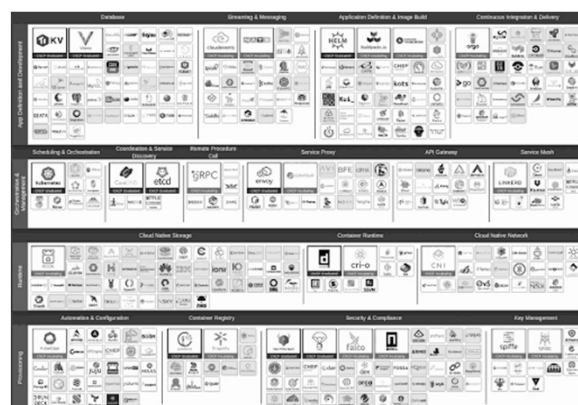


Fig. 2. CNCf Cloud Native Trail Map 2

6

微服務

- 微服務是一種架構式和組織式的軟體開發方法，其中應用程式由小型獨立服務或元件組成，並透過定義良好的 API 進行通訊，服務會一起運作以構成整個應用程式它們是圍繞著業務功能而組織、為鬆散耦合狀態、可獨立建構部署及擴展、並且可測試。
- 微服務可以透過容器化技術將應用程式部署在容器中，並使用 Kubernetes 來有效率管理調度容器，結合 DevOps 與持續整合/持續交付(CI/CD)達到快速開發的目的。
- 在微服務架構下，企業能以小規模的團隊個別管理不同的獨立模組，服務個別部屬，可針對個別新技術獨立更新，使開發流程更快速且有彈性，減少過往為了改一小部分程式碼導致整個應用程式都受影響的狀況出現，獲得更高的生產力。

7

微服務(cont.)

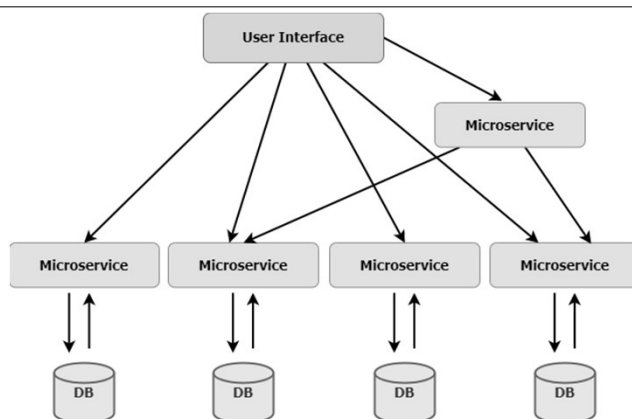


Fig. 3. 微服務架構

8

Kubernetes(K8s)

- Kubernetes是一種容器管理系統，可以用來排定與自動化執行容器化應用程式的部署、管理及擴充，適用於當容器數量增加，需要穩定容器環境，以及管理資源或權限分配的狀況。
- 2021年CNCF調查報告指出，高達96%企業正在使用或是評估採用K8s，逐漸成為主流技術。國內各大科技廠、民間企業及政府機關也都逐步採用K8s、微服務等技術。

9

Kubernetes(K8s) (cont.)

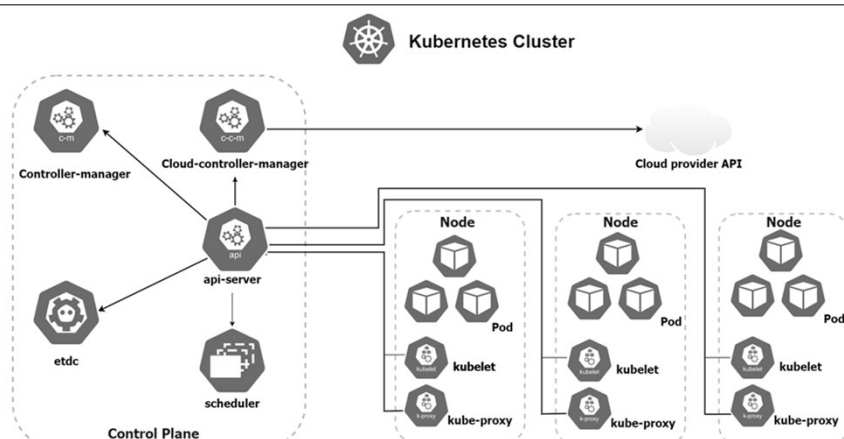


Fig. 4. Kubernetes架構

10

動機

- 企業的資訊系統，常因開發廠商不同，使用不同的建置環境、系統架構及資料庫，而面臨舊系統功能重複、無法單一登入、架構老舊不易維護等問題。在新建系統上也常因現有網路環境與伺服器硬體架構限制、影響線上運作環境、新舊資料整合問題及經費不足等因素，導致困難重重。
- 本研究設計一套智慧管理系統，使用現有資源環境，導入微服務架構，在不影響線上系統運作為前提，將各舊系統的資料整合並利用開發之API將資料以指標的方式呈現，並保留未來新服務應用的開發空間。

11

目的

- 微服務的蓬勃發展，使我們在開發系統時可將服務功能切分，並可個別佈署、更新及測試，新團隊的成員可以更容易瞭解程式碼並迅速做出貢獻。使用Kubernetes為容器管理平台，服務系統部屬更方便，從服務的自動部署、自動修復、負載平衡、權限控管，到服務間的溝通與滾動式更新，都可透過K8s完成。
- 本研究智慧管理系統之設計，以現有網路環境資源為背景，使用既有三台伺服器，擴充記憶體及硬碟空間，並增加既有同型號防火牆與交換器各一台，與既有環境整合為Full HA架構，建置微服務架構，新建置之Oracle資料庫整合既有各系統的資料庫，並透過各種技術組合，開發API程式進行資料交換，呈現各項公司營運指標，管理者能用網頁儀表板觀看並查詢想要的資料，也保留未來新系統的擴充空間，新開發的系統可經由設定，快速使用此架構上的資源，達到快速部屬的目地。

12

2. Research Method

13

研究方法

- 本研究利用分析現有系統資源，進行網路架構設計、系統硬體架構設計、軟體架構設計、系統功能設計，從硬體規劃、系統建置到軟體開發，完成智慧管理系統。
 - 新建置防火牆及交換器，整合現有網路架構，
 - 使用現有硬體伺服器，架設Proxmox VE建立虛擬機叢集，並使用Kubernetes為容器管理調度平台
 - 整合既有前台及外部系統資料庫，以提供指標之運算與應用，
 - 設計標準應用程式介面服務功能，建置一標準API，介接現有系統功能。
 - 建置資料指標監控儀表板管理平台，內容包含指標呈現、統計資訊與服務監控功能。
 - 預留新系統擴充開發空間。

14

網路架構設計

- 增加既有同型號防火牆與交換器各一台，與既有環境整合為Full HA架構，並向外串接各系統介面，並透過防火牆設定NAT隔離避免架構資訊揭露，再加上Policy僅允許必要之存取流量通過，確保資訊安全。
- 針對既有三台HPE伺服器上四個1000Base-T網路埠，以兩埠為一組進行合併（每埠介接至不同網路交換器上）設定成兩組Bond，並配合交換器端設定為LACP或Active/Passive(A/P)運行模式，以提供交換器或單一網路埠故障下的容錯(Fault Tolerance)機制。

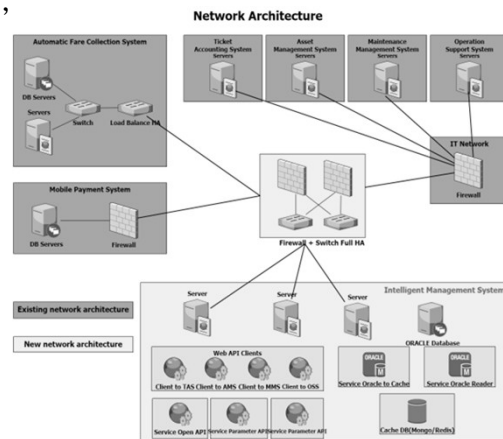


Fig. 5. 網路架構

15

系統硬體架構設計

- 運用現行既有三台HPE伺服器，架設Proxmox Virtual Environment建立Kernel-based Virtual Machine叢集，為安裝Kubernetes(K8s)環境之Linux虛擬機提供高可靠度(High-Availability)和即時遷移(Live Migrate)相關應用服務。

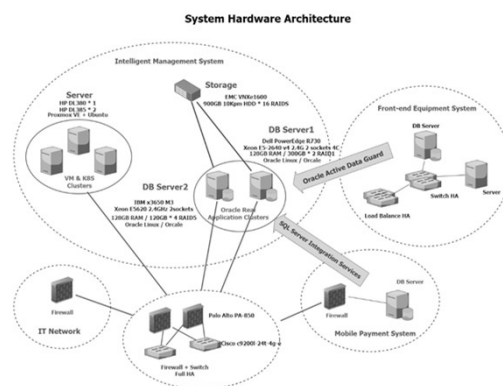


Fig. 6. 系統硬體架構

16

系統硬體架構設計(cont.)

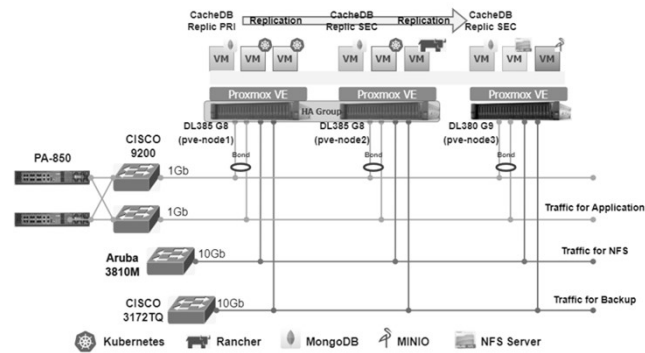


Fig. 7. 虛擬平台系統

17

儲存空間架構

- 將兩台 HPE DL385 G8 伺服器內接硬碟設定為 RAID 1 提供容錯儲存空間。
- 將 HPE DL380 G9 伺服器內接硬碟設定為 RAID 6 以提供容錯儲存空間，並以 NFS export 提供服務。
- NFS export 主要擔任 2 台 HPE DL385 G8 伺服器之網路共享(NFS share)。

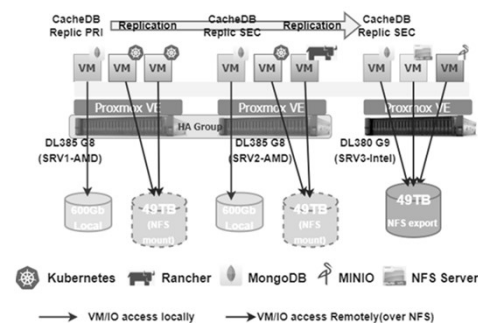


Fig. 8. 儲存空間

18

網路容錯架構

- 針對既有三台HPE伺服器上四個1000Base-T網路埠，其中兩埠為一組進行合併（每埠介接至不同網路交換器上）設定成Bond，並配合交換器端設定為LACP或Active/Passive(A/P)運行模式，以提供交換器或單一網路埠故障下的容錯(Fault Tolerance)機制；擔任Application data traffic之用。

19

虛擬機容錯架構

- 將既有兩台HPE DL385 G8伺服器設定為HA Group。
- HA Group主要針對儲存於網路共享儲存空間(NFS Share)上之K8s環境Linux虛擬機提供高可靠度(High-Availability)機制；當任一HPE DL385 G8伺服器未預警停止服務時，該伺服器上相關之K8s環境Linux虛擬機將被轉移至仍然存活的另一台HPE DL385 G8伺服器上重新啟動運行。
- 針對儲存於三台HPE伺服器本機儲存空間(Local Volume)內提供MongoDB服務之Linux虛擬機，由MongoDB Replica提供高可靠度機制。

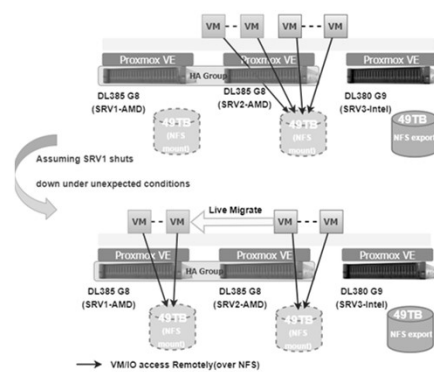


Fig. 9. 虛擬機容錯機制

20

備份機制架構

- 將Backup Client安裝於各Linux虛擬機內，透過原架構提供之本研究獨立VLAN串接至CISCO3172TQ交換器，應用既有Veritas Netbackup 5250 Appliance針對Linux作業系統、Database以及Application執行備份及還原作業。

21

軟體功能架構設計

- 軟體建置微服務之容器所採用作業系統預計預設為Ubuntu 20.04以上版本，資料庫則將採用MongoDB /Redis(NoSQL)及Oracle。整體架構使用前後端分離，核心程式部分預計採用.Net 6/EF Core 6，前端網頁預計使用Vue.js開發。
- 本系統軟體架構圖及系統資料流程圖如下：

22

資料庫同步

- Oracle RAC

RAC 節點會通過高速交互連接而連接起來，以便在 Oracle 節點之間進行高速通訊。這些節點可以在啟動期間交換各種資料區塊所有權資訊、鎖定資訊、交換交易資訊和資料。

- Oracle ADG

系統是一種具有高可用性的災變復原(Disaster Recovery, DR)解決方案，由一個主要資料庫(Primary Database)提供服務，且同時維護並監控一個或多個備援資料庫(Standby Database)，備援資料庫透過持續更新 Redo 資訊與主要資料庫保持同步。

- SQL to Oracle

25

系統功能說明

- 本系統提供各關鍵指標運算，以利有效且即時整合子系統相關資料。系統主要項目含：
- 1. API服務：與內/外部系統(TAS系統、AMS系統及MMS系統)資料串接入口，並具備告警資訊推播給外部系統。
- 2. 服務指標運算：計算報表、告警資訊及計算外部系統所需資料，資料來源為 Oracle 資料庫及 API 服務，查詢資料儲存需求需建立快取資料庫。
- 3. 監控儀表版：使用者圖形化介面(UI)供使用者操作，且具備帳號管理、報表產出、資料查詢及告警/監控項目設定。

26

3. Experimental Results

27

智慧管理系統畫面呈現

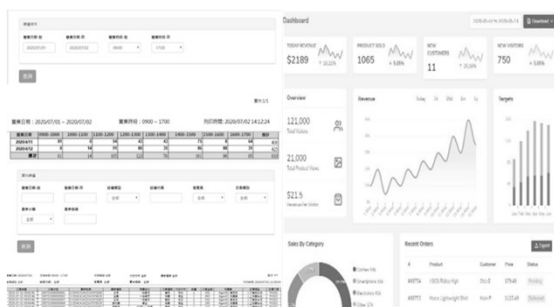


Fig. 12. 新舊系統畫面比對



Fig. 13. 系統畫面呈現

28

系統監控與告警

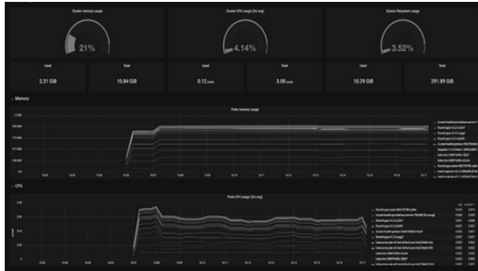


Fig. 12. Grafana儀表板

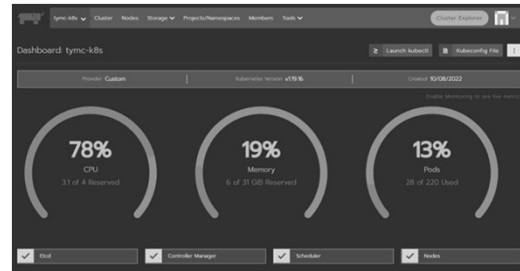


Fig. 12. K8s資源監控

4. Conclusion

結論

- 資訊系統更是企業營運重要的核心，如何有效率的應用資料是企業成長的關鍵。隨著近年來容器技術及容器平台(K8s)的成熟，微服務架構所帶來的開發與部署程式的好處，成為舊系統汰換升級、新系統建置的主流應用技術。
- 本研究設計之智慧管理系統，除了提供從既有系統蒐集的資料整合，以各種關鍵指標以視覺化呈現，讓公司管理者作為有效營運分析參考使用之外，同時也藉此建置微服務架構，從網路環境、硬體架構的規劃設計到資料整合的API開發設計，並保留未來新服務應用的開發空間，可供位於數位轉型階段的企業做參考。

31

簡報完畢
謝謝聆聽

32