

在服務導向物聯網下以服務需求管理作業 提升系統服務效能---以防災監測應用為例

王順生
朝陽科技大學
工業工程與管理系副教授
sswang@cyut.edu.tw

王淑卿*
朝陽科技大學
資訊管理系教授
scwang@cyut.edu.tw

嚴國慶*
朝陽科技大學
企業管理系教授
kqyan@cyut.edu.tw

王信傑
朝陽科技大學
資訊管理系研究生
s10014609@cyut.edu.tw

*：聯絡人

摘要

物聯網(Internet of Things ;IoT)的發展，使得現實生活中各種智能物件(Smart Object)之間能夠相互溝通，透過網際網路的應用，以及不同類型的感測器(Sensor)與物件的結合，可促進多元化的應用程序發展，如智慧城市、智慧健康照護和智慧居家等。物聯網的運用，讓使用者可遠端操控生活中的各種物件，享受物聯網帶來的服務。然而，大量資訊的流通都透過物聯網相連結時，為了提供大量的使用者對物聯網不同應用程序的服務需求，雲端運算環境的結合將是物聯網發展的一個重要環境。因此，本研究提出在服務導向物聯網下建構服務需求處理作業，透過雲端運算環境進行大量服務需求處理，提升物聯網應用之系統服務效能和提供差異化服務，滿足大量使用者對物聯網的服務需求。

關鍵詞：物聯網、服務導向架構、CPU 排班演算法、排程演算法。

Abstract

Due to the development of the Internet of Things (IoT), the smart objects can communicate with each other in the real world. Through the internet and combination of different types of sensors, IoT can promote the development of applications, such as intelligent city, intelligent healthcare and smart home. In the IoT, it is convenience to enjoy the service of remotely to operate the things in the life. However, all the information is interconnecting in the IoT, how to provide the services of IoT, the combination of cloud computing is very important to promote the development of IoT. Thus, the service management operating processes of IoTSoA (Service Oriented Architectures underlying IoT) is constructed in the study that can enhance the

quality of service of the IoT's application, to provide differentiated services and meet the needs of users.

Keywords: Internet of Things, Service-oriented Architecture, Cloud Computing, Scheduling.

1. 前言

物聯網最早源自於 RFID (Radio Frequency Identification)的概念，並於 1999 年由 Kevin Ashton 所提出[1]，而後於 2005 年由國際電信聯盟(International Telecommunication Union)正式提出物聯網的概念[7]，使得物聯網的發展不再侷限於 RFID 的技術範圍。物聯網企圖將全世界連結成一個虛擬可循環的網路，透過網際網路的結合，能夠將現實世界中的物件相互連接，形成一個虛擬的網路世界[4,9,14]。物聯網的概念，使得人與物件之間，甚至物件與物件之間能夠相互進行溝通，透過網際網路與各種不同無線通訊感測網路的結合運用，如 RFID、Zigbee、WiFi 及 UMTS(Universal Mobile Telecommunications System)等，這些物件能夠遠距離進行操作及資料交換。在物聯網的環境中，透過各種感測器與物件的結合，能夠賦與人們的日常生活中的物件具有運算能力、判斷能力及環境監測能力等，使得普及運算環境(Ubiquitous Computing)來臨，讓人們的生活變得更智慧化。

物聯網的概念可促進多元化的應用程序發展，如防災監測、智慧城市、智慧健康照護和智慧居家等。透過物聯網的應用服務端，運用大量的智慧設備及網際網路的結合，人們可以遠端享受物聯網所帶來的便利性[4]。然而，當多元化的應用程序透過物聯網整合在一起並提供服務時，不同應用程序之間的使用者所發出的服務需求，將為物聯網帶來大量的服務工作需求。此外，各種應用程序之間的服務需求工作不一。以防災監測為例，如颱風監測應用

程序可提供風速偵測服務、降雨量偵測等服務；而土石流監測應用程序則提供降雨量監測服務、土石鬆動監測等服務。這些不同應用程序之間的服務需求工作可能互為異質或同質。且當大量的服務需求同時流入物聯網時，若未有一個良好的機制及環境來進行大量服務需求的處理，大量且複雜的服務需求工作將可能影響物聯網相關應用程序的服務品質，使得使用者在享受物聯網的服務受到影響。

因此，本研究在服務導向物聯網下提出服務需求管理作業。研究中將運用雲端運算環境進行大量服務需求處理，及針對不同應用程序的服務需求工作提供不同的排程，使大量且異質的服務需求工作能夠在雲端運算環境中被有效率的處理，藉以提供應用程序端使用者所需要的服務，避免工作受到壅塞。

本文第 2 節為文獻探討，將說明雲端運算、CPU 排班演算法及排程演算法；第 3 節說明本研究所使用的以服務為導向之物聯網架構；第 4 節為本研究所提出的服務需求管理作業；第 5 節為範例說明；最後一節則是結論及未來的工作。

2. 文獻探討

本節將說明雲端運算、CPU 排班演算法及排程演算法

2.1 雲端運算

雲端運算利用了分散式系統(Distributed System)的特性，藉由網際網路的使用，透過匯集不同效能的電腦，使得多台低成本的電腦能夠進行大量的運算工作處理，來達到如同超級電腦的處理能力。雲端運算採用叢集式(Cluster)的運算使各個區域的電腦能夠相互運用，以形成分散式的運算[5]。雲端運算可藉由叢集式架構分配適當的節點來服務使用者或應用程序的需求，以提供多元化的服務需求，如儲存空間、運算能力或其他服務等。此外，雲端運算運用了資源虛擬化及動態可擴充資源的技術，將使得雲端運算環境中的資源能更有效的被運用[5,12]，讓公用運算的概念能夠讓企業按需求租賃所需的資源，即時滿足企業的需求。

在雲端運算的服務中，主要可區分為三大項[10,12]，分別為軟體即服務(SaaS, Software as a Service)、平台即服務(PaaS, Platform as a Service)、及基礎設施即服務(IaaS, Infrastructure

as a Service)。軟體即服務可提供軟硬體資源，形成多租戶的平台，讓雲端使用者只需連上網際網路，不需耗費額外的硬體設施成本，只需要透過租賃的方式，即可享受到軟體的服務。平台即服務可提供雲端使用者在網際網路上開發各種應用軟體所需的系統軟體和環境，使用者只需運用網際網路連上雲端，即可享受開發系統的服務。基礎設施即服務可提供各種硬體設施的租賃，讓雲端使用者無須耗費建置硬體設備的時間及成本，高靈活與可擴充性的方式，讓使用者只需依據本身需求，向雲端供應商租賃硬體設備即可。

整體而言，雲端運算不是一種新興的運算方式，而是以分散式運算的概念為基礎延伸而來[5,8,12]，雲端運算描述了一種基於網際網路及資訊技術所提供的新型服務、使用和交付模式，通常涉及透過網際網路來提供動態易擴充功能，而且經常是虛擬化的資源[8,10,15]。典型的雲端運算供應商能夠提供通用的網路應用服務，讓使用者可以透過瀏覽器等軟體或者其他 Web 服務來存取儲存在伺服器上的應用程序和資料[11][13]。

在物聯網環境下導入雲端運算平台，利用分散式系統的特性建立和超級電腦一樣的運算能力，並且運用公用運算的概念，讓企業依據實際的需求向雲端供應商租賃計算能力、儲存空間和網路流量服務等。整體而言，雲端運算提供網路上的資源來執行使用者端的需求，以提升資料處理的速度，使企業經由雲端運算平台處理物聯網所帶來的海量資料並快速建立出物聯網資料處理的環境。因此，本研究將運用雲端運算環境的特性建立物聯網的基礎架構。

2.2 CPU 排班演算法

CPU 排班演算法為運用在單一處理器中的排班計算，在單一處理器的佇列中同時具有多個不同的程序需要被執行時，因此將挑選一個合適的程序將任務執行。在 CPU 執行程序中，當具有高優先權的任務進入時，為了使高優先權的任務快速被執行，在執行序之間存在著可搶先(Preemptive)的機制，使得高優先權任務可即時取得 CPU 的運算服務，並且將執行中的任務轉變為等待的狀態。而不可搶先(Non-preemptive)的機制，使得工作一旦進入 CPU 執行序中被服務，直到工作完成之後，才能繼續服務新的工作。在本小節中，將針對常

用的 CPU 排班演算法進行說明[1]。

FCFS(First Come First Served)：先進先出排班演算法是根據程序的抵達時間作為排序的順序且具有不可搶先的特性，越早抵達的程序將會優先的進行排班。因此，執行時間較短的程序可能受到執行時間較長的程序所影響，而造成程序的等待時間(Waiting Time)較長。

Round-Robin(RR)：RR 排班演算法為一種公平性的排班演算法，以輪巡式的方式，使每個程序能公平地獲得固定服務額度而被執行。然而在公平性的原則下，RR 無法提供差異化的服務。且當服務額度設定過高時，其效能將趨近於 FCFS；若設定過低，將導致任務交換次數過高，影響排班作業。

SJF(Shortest Job First)：SJF 排班演算法短程序優先排班演算法為一種不公平性的排班演算法，程序在 CPU 執行時，以最短完成時間作為排序順序，程序的執行完成時間越小越優先被執行，而具有相同完成時間的程序，則依照先進先出排班演算法進行排序。短程序優先排班演算法的好處，是讓程序之間的等待時間不會過長，能夠獲得較小的平均等待時間。然而短程序優先排班演算法是以高優先權程序優先執行，與優先權排班演算法一樣為不公平性的排班方式，最大的問題可能造成程序產生飢餓的問題。此外當較大的程序需要被優先執行時，短程序優先排班演算法無法彈性的調度程序。

E-SJF(Enhanced SJF)：E-SJF 排班演算法是 SJF 排班演算法的改良，由王淑卿等學者所提出[1]，為了解決 SJF 排班演算法所造成的長程序執行時間被延遲過久的問題。E-SJF 以最長程序的執行時間複雜度為門檻值做為搶先排班的依據，當已被執行的程序的總和時間超出門檻值，運用搶先機制讓最長執行時間的程序得以執行，減少等待時間過久的問題。然而 E-SJF 排班演算法無法彈性的針對程序需求彈性調度其執行順序，雖能減少長程序的延遲時間，但無法即時讓需要被執行的程序優先執行。為能適用於物聯網的程序排班，在本研究中，提出 WE-SJF(Weight Enhanced SJF)排班演算法進行 E-SJF 排班演算法的改良。

2.3 排程演算法

多程序的工作處理中，工作的處理速度間

接影響對使用者的服務，為了提高系統執行的效率，透過排程演算法能使資源被有效的運用，因此選擇適合的排程演算法，將影響系統執行的效能[6]。在本小節中，將針對常用的排程演算法進行說明。

Min-Min：Min-Min 排程演算法在預先規劃的計算成本內，針對每一個未被執行工作的排程，建立最小的工作完成時間，並分派工作至可提供最小完成時間的節點中服務，可使工作與節點皆獲得最小的工作完成時間，因此稱之為 Min-Min 排程演算法。

Max-Min：Max-Min 排程演算法則在預先規劃的計算成本內，針對每一個未被執行工作的排程，建立最小的工作完成時間，而後挑選工作於最大完成時間的節點中接受服務，因此稱之為 Max-Min 排程演算法。由於 Max-Min 排程演算法為挑選最大的完成時間之工作節點優先進行服務，以致無法獲得較低的工作完成時間。

3. 以服務為導向之物聯網架構

在王順生等學者的研究中提出「在物聯網建構以服務為導向之架構(SoA underlying IoT)」稱為 IoTSoA。IoTSoA 將物聯網分成四大層級，分別為感測層、中繼管理層、雲端管理層及應用服務層，如圖 1 所示[2]。

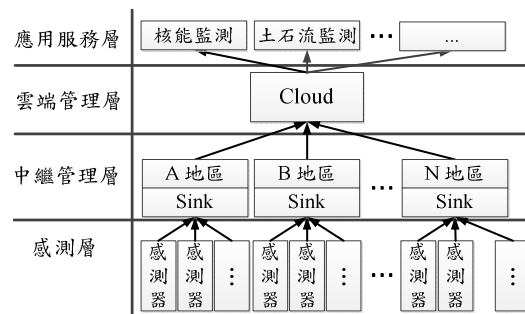


圖 1 IoTSoA [2]

IoTSoA 最底層為感測層，主要的工作為收集各種感測資料，以提供應用端多元化的應用服務。這些感測資料的蒐集，將依據不同的感測器的功能而不同，如溫度、濕度、位置。而由於一般感測器的能力通常有所限制，如儲存、運算處理能力、傳輸能力、及電源等。因此，在感測器能力有限的情況下，為了傳輸且蒐集物聯網中龐大的感測資料，單靠感測器是

無法負荷這項工作，所以在感測層所蒐集的資料，將持續傳輸到 Sink 端，透過各地區的 Sink 端，作為資料傳輸的中繼站。

IoTSOA 的第二層為中繼管理層。中繼管理層是由位於各個不同區域的 Sink 端所構成，其持續接收來自感測層的資料，將感測資料進行資料彙集、壓縮及儲存，並將感測層的資料傳輸至雲端管理層中。

IoTSOA 的第三層為雲端管理層。雲端管理層的作業主要區分為兩項，第一項為持續接收來自中繼管理層的感測資料，並針對來自不同地區 Sink 端的感測資料進行相對應的資料處理機制，以及匯集資料至雲端管理層中進行儲存。第二項則根據不同使用者所發出的服務需求，將相對應的感測資料轉化為各種服務送至應用程序中，使得物聯網能夠提供多元化的應用程序服務給使用者。在本研究中，將探討在物聯網中，為了轉化大量的物聯網資料並提供服務給使用者，當雲端管理層收到大量的服務需求工作時，將能夠針對不同應用程序及服務種類，使雲端節點能夠服務物聯網中大量且可能為異質的服務需求工作，使後續的應用服務層能正常且提出足夠服務品質的執行。

IoTSOA 的第四層為應用服務層。將根據不同使用者的需求，發展出各種物聯網的應用，以滿足使用者的需求。當大量的使用者透過應用端提出不同的服務需求時，應用程序將提取雲端管理層中所儲存的感測資訊，以提供使用者相對應的資訊服務。

4. 在服務為導向物聯網架構下之服務需求管理作業

由於物聯網中大量的使用者發出的服務需求若同時在物聯網上流通時，如果未有一個能容忍大量服務需求處理的環境，及有效率進行服務需求工作處理，將可能造成工作處理的壅塞。為了能在物聯網上快速地提供來自應用端使用者的服務需求，因此本研究在 IoTSOA 的架構，於雲端管理層中運用雲端運算的平行運算特性，藉由本研究所提出物聯網服務需求管理作業，以因應來自物聯網中不同應用程序之間的使用者需求，使物聯網能夠有效率地服務大量的使用者，以減少服務需求工作處理的壅塞。此外，研究中更考量使用者間可能具有不同的服務品質，因此針對不同工作提供差異化的服務機制。服務需求管理作業分為五個元件，分別為應用程序服務分類代理人、應用程

序服務佇列、服務需求雲端資源代理人和服務管理雲端節點選派代理人。服務需求管理作業執行流程如圖 2 所示。以下小節將分別介紹服務需求管理作業中的各部分。

4.1 應用程序服務分類代理人

應用程序服務分類代理人的主要工作，為針對各種來自不同應用程序的服務需求進行分類。在本研究中，來自各種應用程序端的服務需求，透過網際網路享受物聯網服務時，其服務的內容將依據各個應用程序有所不同，如土石流防災監測、颱風防災監測和火災防災監測等。為了避免物聯網在服務多元化的應用程序時，大量的服務需求造成阻塞，且能夠服務各個應用程序的異質工作，因此透過雲端運算環境的平行運算特性，運用應用程序服務分類代理人先將服務需求依其工作性質進行區分。

換言之，在物聯網環境中當大量服務需求發送到雲端時，為了處理的來自不同應用程序的服務需求，物聯網在處理服務需求時，首先透過應用程序服務分類代理人依據不同來源的應用程序服務需求，而分派到各個應用程序服務佇列。接著，運用雲端的平行運算特性，使得來自多元化應用程序的大量異質任務能夠分類到相對應的應用程序佇列中進行排班。

4.2 應用程序服務佇列

應用程序服務佇列是由多元化的應用程序所組成，為了能夠有效率的執行物聯網中的服務需求工作及進行工作的排班，透過設置各個不同的應用程序服務佇列，使得服務需求工作能夠在屬於自己的應用程序服務佇列進行排班，簡化異質工作的複雜性。

在本研究的應用程序服務佇列中，為了提供物聯網使用者差異化的服務，當大量的服務需求進入時，考量使用者之間不同等級的服務機制，使得各應用程序可依條件設置工作處理的優先權規則，例如有付費的使用者，具有較高優先權以取得物聯網服務；或使用者在滿足特定條件的情況下，能夠獲得較快速度的物聯網服務等。因此，使用者將依據不同的權限進行任務排班，以提供差異化的處理速度，滿足使用者的不同權限之需求。

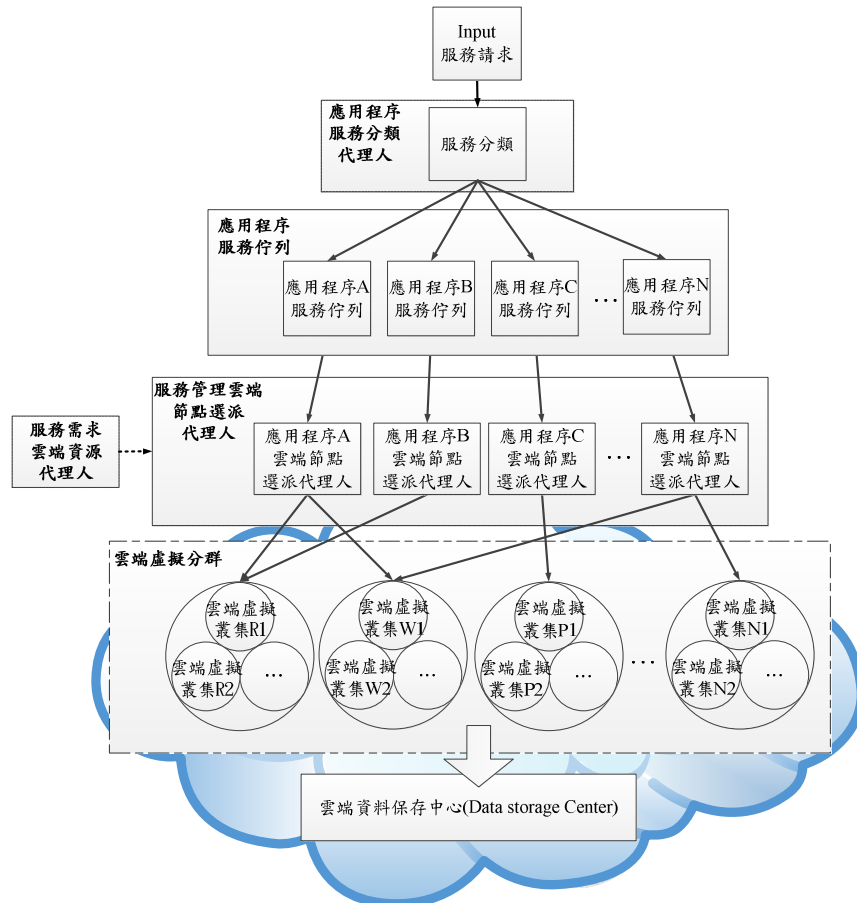


圖 2 服務需求管理作業執行流程

為了因應不同權限的需求，且能夠大量服務物聯網任務，本研究提出 WE-SJF(Weight Enhanced SJF)排班演算法以進行工作的排班。利用 WE-SJF 排班演算法的權重區分特性，使任務能夠依據不同的應用而區分等級，WE-SJF 針對等級越高的任務優先進行排班，而當任務的 Weight 值相同時，將具有 SJF 排班演算法的特性，使得任務獲得較小的任務等待時間(Waiting Time)。其中，Weight 的設定將根據不同應用程式而設定改變。

以下為 WE-SJF 排班演算法的計算過程：初始 Weight 值則設置為 1。WE-SJF 排班演算法，可分為 7 個步驟：

[步驟 1]：將佇列中的任務，分別依據各自的 Weight 值進行排序，使任務的排班能依據 Weight 值由高至低排班。

[步驟 2]：根據 Weight 值的不同輪循將各個等級 Weight 值中的等待任務執行時間由小到大排序。

[步驟 3]：挑選佇列中最長的任務執行需求，

設為最大容忍執行門檻值。

[步驟 4]：依據[步驟 2]之排序，挑選任務執行需求最小的任務執行排班。

[步驟 5]：將輸出之任務依據已輸出的任務執行需求累計至輸出門檻值。

[步驟 6]：

(1) 當累積的輸出門檻值超出最大容忍執行門檻值時，將佇列區中最長執行需求的任務搶先排班輸出。

(2) 重新挑選最長任務執行需求設為最大的容忍執行門檻值。

[步驟 7]：重複執行[步驟 1]至[步驟 7]，直到佇列的任務分派完畢。

4.3 服務需求雲端資源代理人

服務需求雲端資源代理人的主要工作為蒐集各雲端節點的工作負載情形，以及制定應用服務的服務需求處理群集表。為了能夠服務大量的物聯網服務需求，因此運用雲端運算的分散式運算特性，藉由使用大量雲端節點，來

進行大量工作的服務。考量在雲端節點中，工作的等待時間越高，表示該節點可能已經負載過多的工作程序或正在執行需求較大的工作，將可能影響工作處理的服務品質。且雲端運算節點在執行服務需求工作時，可能根據不同的服務需求而提供不同的工作處理機制，如土石流防災監測應用可提供降雨量資料查詢服務、淺勢溪流查詢服務、土石流警戒地區查詢服務等；而颱風防災監測應用可提供降雨量資料查詢服務、風速資料查詢服務、颱風路徑查詢服務等。在上述的應用程序服務中，不同應用程序之間可能具有相同類型的服務工作，如降雨量資料查詢服務。而本研究則以服務需求工作類型的不同而虛擬分群雲端叢集，藉由多個不同的群集設置，讓來自不同應用程序的服務需求工作，能夠分派到符合該筆工作可被進行處理的雲端節點中進行服務。換言之，在雲端群集配置上，將依據各種應用程序中的服務需求工作類型，而設置不同的虛擬群集。

因此，本研究將建立服務需求處理群集表，如表 1 所示為例。考量不同應用程序之間可能提供相同的服務，因此建立其服務群集的關係。此外考量雲端運算單一節點可能因負載量過高而造成任務的延遲，因此建立服務處理雲端節點負載量表。服務管理雲端節點選派代理人可透過服務處理雲端節點負載量表，使工作能夠轉換至其它叢集節點進行服務，讓雲端節點能夠處理大量的服務需求工作。

表 1 服務需求處理群集範例

應用程序	可提供服務工作	工作可被服務群集
土石流 防災監測	降雨量查詢	叢集 R
	淺勢溪流查詢	叢集 W
颱風 防災監測	風速查詢	叢集 Y
	降雨量查詢	叢集 R

4.4 服務管理雲端節點選派代理人

當大量的服務需求工作需要分派雲端環境的各個節點進行處理時，本研究透過服務管理雲端節點選派代理人將工作選派到合適的節點中進行服務，使大量的服務需求能夠有效率地被雲端節點服務。本研究運用 Min-Min 排程演算法的特性，將目前未排程的工作，分別建立最小的工作完成時間，並將工作分派給最小工作完成時間的節點處理，而得到較小的工

作等待時間。

本研究所使用的 Min-Min 排程演算法包含三個步驟：

[步驟 1]:各服務管理雲端節點選派代理人計算目前等待執行的工作，在各個叢集雲端節點所需的工作完成時間。

[步驟 2]:挑選於雲端節點，執行工作所需最短完成時間的節點，將工作分派至該節點中服務，並等待正在執行中的工作結束。

[步驟 3]:重複執行[步驟 1]至[步驟 3]，直到佇列中的工作執行完畢。

5. 範例說明

在本節中將以在 IoTSoA 架構中的「防災監測服務」實際應用為例，說明本研究所提出的服務需求管理作業確實能夠有效率的進行大量服務需求工作處理，及提供多元化應用程序服務物給使用者。以下實例為服務需求管理作業的執行過程四步驟。

【步驟一】：應用程序服務分類代理人

假設分別收到土石流監測應用程序(DFM)和颱風監測(TM)應用程序的服務需求。應用程序服務分類代理人將任務分類至相對的應用程序中進行任務排班，表 2 為服務需求任務分類結果。服務需求的種類可區分為 a、b、c、d、e，分別表示 a 為降雨量資料查詢、b 為淺勢溪流資料查詢、c 為土石流警戒地區查詢、d 為風速資料查詢和 e 為氣壓資料查詢。

雲端管理層所收到的服務需求任務都各具有 Weight 值，而 Weight 值的變化將影響任務執行順序。在實際應用時，可依據不同應用的程序需求而設置變化條件，而本研究以土石流防災監測為例，將判斷使用者的所在位置是不是位於土石流發生次數較高的區域而給予較高的 Weight 值。

表 2 服務需求任務分類結果

土石流防災監測 應用佇列	颱風防災監測 應用佇列
DFMS1-a	TMS1-d
DFMS1-b	TMS1-a
DFMS2-a	TMS2-d
DFMS3-a	TMS1-e
DFMS1-c	TMS3-d
DFMS4-a	TMS2-a
DFMS2-c	TMS2-e

【步驟二】：應用程序服務佇列

運用 WE-SJF 排班演算法在各應用程序服務物佇列中進行排班計算，以土石流防災監測為例進行排班。表 3 為土石流防災監測佇列中任務的執行需求時間。表 4 為土石流防災監測佇列排班結果。

表 3 土石流任務的執行需求時間

任務	執行需求時間	Weight
DFMS1-a	56	2
DFMS1-b	89	1
DFMS2-a	100	1
DFMS3-a	80	2
DFMS1-c	72	1
DFMS4-a	38	1
DFMS2-c	18	2

表 4 土石流防災監測佇列排班結果

執行順序	執行需求時間	Weight
1	DFMS2-c	2
2	DFMS1-a	2
3	DFMS3-a	2
4	DFMS4-a	1
5	DFMS1-c	1
6	DFMS2-a	1
7	DFMS1-b	1

【步驟三】：服務管理雲端節點選派代理人

各個應用程序的服務管理雲端節點選派代理人，運用 Min-Min 排程演算法進行工作分派，並對照服務需求處理群集表，依據服務的種類將工作分派到對應群集中進行處理，本研究以土石流監測應用程序的 a 類型工作為例進行工作分派。表 5 為服務需求處理群集，依據服務需求工作的類型而虛擬分群雲端叢集，使得工作將分派到特定的群集中進行服務需求工作處理。表 6、表 7、表 8 為土石流服務需求工作分派至 SA 叢集中，3 個叢集節點工作分派前所需的工作完成時間。

經由 Min-Min 排程演算法的計算，將分派工作至節點中能夠獲得最短的工作完成時間進行工作處理。因此 a 類型的服務需求工作的總完成時間為 35。

【步驟四】：服務管理雲端節點選派代理人

服務需求雲端資源代理人將更新各個叢集的負載情形，以提供各雲端節點應用選派代理人最新的節點負載狀況。表 10 為 SA 群集中的各叢集節點負載狀況。

表 5 服務需求處理群集表

服務管理雲端節點選派代理人	工作類型	工作可被服務之群集	各群集中的叢集
土石流監測選派代理人	a	SA	SA1, SA2, ...
	b	SB	SB1, SB2, ...
	c	SC	SC1, SC2, ...
颱風監測選派代理人	a	SA	SA1, SA2, ...
	d	SD	SD1, SD2, ...
	e	SE	SE1, SE2, ...
...

表 6 SA1 叢集工作分派前所需的完成時間

節點 工作	SA1-1	SA1-2	SA1-3	SA1-4
DFMS1-a	48	54	64	41
DFMS3-a	32	33	104	82
DFMS4-a	49	12	71	70
DFMS2-a	23	37	87	103
TMS1-a	57	59	56	42
TMS2-a	15	43	16	93

表 7 SA2 叢集工作分派前所需的完成時間

節點 工作	SA2-1	SA2-2	SA2-3	SA2-4
DFMS1-a	81	27	72	64
DFMS3-a	10	40	72	76
DFMS4-a	75	67	83	29
DFMS2-a	25	100	83	64
TMS1-a	64	61	47	69
TMS2-a	97	95	21	19

表 8 SA3 叢集工作分派前所需的完成時間

節點 工作	SA3-1	SA3-2	SA3-3	SA3-4
DFMS1-a	57	85	19	64
DFMS3-a	38	27	83	96
DFMS4-a	93	48	97	89
DFMS2-a	97	108	95	42
TMS1-a	18	85	79	51
TMS2-a	78	46	75	59

表 9 土石流監測選派代理人工作排程結果

任務	節點	Min-Time
DFMS3-a	SA2-1	10
DFMS4-a	SA1-2	12
TMS2-a	SA1-1	15
TMS1-a	SA3-1	18
DFMS1-a	SA3-3	19
DFMS2-a	SA2-1	35

表 10 SA 群集中各叢集節點的負載狀況

叢集(i) 節點(j)	SA1	A2	A3	A4
1	+15	+35	+18	
2	+12			
3			+19	
4				

6. 結論及未來工作

在物聯網的環境下，運用了大量的感測資料發展出多元化的應用程序，讓使用者透過國際網路即可遠端享受物聯網所帶來的便利服務。然而當這些來自多元化的應用程序使用者，在同一時間向物聯網發出服務需求時，如何因應物聯網中大量且異質的服務需求，並維持物聯網的服務品質及提供差異化服務，為本研究考量的重點。

因此本研究在以服務為導向物聯網架構 (IoTSoA) 下提出服務需求管理作業，達到下列 4 項目的：1) 運用雲端環境的分散式計算運算特性，使大量的服務需求工作能夠被平行處理；2) 在應用程序服務佇列中進行工作排班，提供工作差異化服務；3) 在服務管理雲端節點選派代理人中進行工作排程及分派，使異質服務需求工作能夠有效率且分派到合適的節點中進行處理；4) 在物聯網以服務為導向架構下能夠提供多元化的應用程序服務給使用者。

在未來的研究中，將在服務導向物聯網架構下探討在雲端管理層中，針對各種應用程序服務提供更適合的資料服務排程演算法，使每個子服務應用節點佇列能夠更有效率地進行任務的排程，並且加入更多的排程因子，如節點的工作能力與影響工作處理的因子，甚至更將考慮節點間的負載平衡，使得物聯網能夠更有效率地處理大量服務需求工作。

致謝

這篇論文是國科會計畫 (NSC101-2221-E-324-032 與 102-2221-E-324-008) 研究成果的一部份，在此我們感謝國科會經費支持這個計畫的研究。

參考文獻

- [1] 王淑卿、嚴國慶、王順生、陳慶維，”在雲端運算環境中以三階段排程機制提升系統服務效能”，*第十六屆資訊管理暨實務研討會*，pp. 2265-2281，2010。
- [2] 王順生、王淑卿、嚴國慶、王信傑，”在物聯網建構以服務為導向的架構和作業流程--以防災監測服務為例”，*第七屆資訊科技國際研討會*，pp. 102, 2013.
- [3] Ashton, K., “That 'Internet of Things' Thing,” *RFID Journal*,” pp. 22, 2009.
- [4] Atzori, L., Iera, A. and Morabito, G., “The Internet of Things: A Survey,” *Computer Networks*, Vol. 54, No. 15, pp. 2787-2805, 2010.
- [5] Aymerich, F.M., Fenu, G. and Surcis, S., “An Approach to A Cloud Computing Network,” *Proceedings of the First International Conference on the Applications of Digital Information and Web Technologies*, pp. 113-118, 2008.
- [6] Braun, T.D., Siegel, H.J., Beck, N., Bölönid, Muthucumar Maheswaran, L.L., Albert, I.R., Robertson, J.P., Theys, M.D., Yao, B., Hensgen, D., and Freund, R.F., “A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems,” *Journal of Parallel and Distributed Computing*, Vol. 61, Issue 6, pp. 810-837, 2001.
- [7] ITU Internet Reports, “The Internet of Things,” 2005.
- [8] Grossman, R.L., “The Case for Cloud Computing,” *IT Professional*, Vol. 11, No. 2, pp. 23-27, 2009.
- [9] Jara, A.J., Zamora, M.A. and Skarmeta, A., “Glowbal IP: An adaptive and transparent IPv6 integration in the Internet of Things,” *Proceedings of the Mobile Information Systems*, Vol. 8, No. 3, pp. 177-197, 2012.
- [10] Luo, Y., “Network I/O Virtualization for Cloud Computing,” *IT Professional*, Vol.

- 12, No. 5, pp. 36-41, 2010.
- [11] Mathur, P. and Nishchal, N., "Cloud Computing New Challenge to the Entire Computer Industry," *Parallel Distributed and Grid Computing (PDGC)*, pp. 232-228, 2010.
- [12] Rimal, B.P., Choi, E. and Lumb, I., "A Taxonomy and Survey of Cloud Computing," *Proceedings of The NCM2009 5th International Joint Conference on INC, IMS and IDC*, pp. 44-51, 2009.
- [13] Vouk, M.A., "Cloud Computing- Issues, Research and Implementations," *Information Technology Interfaces*, pp. 31-40, 2008.
- [14] Welbourne, E., Battle, L., Cole, G., Gould, K., Rector, K., Raymer, S., Balazinska, M. and Borriello, G., "Building the Internet of Things Using RFID: The RFID Ecosystem Experience," *Proceedings of the IEEE Internet Computing*, Vol. 13, No. 3, pp. 48-55, 2009.
- [15] Zhang, S., Zhang, S., Chen, X. and Huo, X., "Cloud Computing Research and Development Trend," *Proceedings of the Second International Conference on Future Networks*, pp. 93-97, 2010.