

# 行動資料格網架構及其複本排程演算法

張玉山

國立台北大學資訊工程系  
ysc@mail.ntpu.edu.tw

鄒國界

國立台北大學通訊工程研究所  
rock00777@msn.com

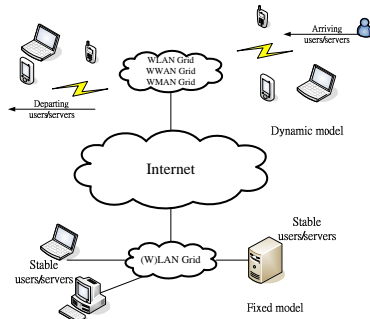
## 摘要

資料格網(data grid)在資料密集的應用中經常扮演重要的角色,近年來行動資料格網的研究已經受到重視。行動資料格網中必需考慮行動設備有限的儲電量能、提供高效率及可靠的機制來存取格網中的資料,由於目前資料格網中的資料複本排程方法並未考慮到行動裝置在無線網路上的各項特性。在本論文中我們利用行動代理人的特性及優點設計一個以行動代理人為基礎的行動資料格網架構,並且在此架構下提出資料複本排程演算法,在此架構及演算法上,用戶端行動代理人與格網系統溝通取得複本資訊,應用我們所提出的複本傳輸排程演算法達到 1. 高效率的資料傳送。2. 整體網路存活率提高,提高行動資料格網的可用性。3. 提高行動資料格網服務的可靠度。

關鍵詞: 資料格網、行動資料格網、格網計算、複本排程演算法

## 一、緒論

由於無線技術與設備的快速發展,行動格網(mobile grid)已經受到重視[3-5, 10-12, 15]並提出了新的問題及挑戰。然而,在無線網路的環境中,當行動裝置移動時,可能會造成網路頻寬與資料傳輸效能的改變,及間歇性地進入或離開格網系統,如圖一所示。另外,行動裝置具有有限的處理電能、有限的儲存空間及計算能力[3-5]等特徵,這些問題都需要一個好的管理機制及方法。



圖一、行動環境之架構

行動資料格網架構[19, 20]必須能提供高效率且可靠的機制來處理在無線網路上高間斷性的網路連線與高網路延遲性,以及高斷線

率可能造成的錯誤,並且必須考慮到每一個行動設備有限的儲電量。目前的資料格網在資料複本排程策略[6, 7, 8]上都考慮傳統的網際網路環境,並未考慮到無線網路上的各項特性,這些方法運用到無線環境,會有一些問題。

行動代理人[23]技術已經成功並廣泛的被運用到各種分散及行動的應用上,像是並行處理、網路監控、通知與分析、安全的中介、分散式資訊的擷取等應用[24]上。運用行動代理人技術具有多項優點[24],像是動態的適應、異質性、獨立且非同步的執行、強健與容錯、具封裝的協定及減少網路傳遞時間等。這些優點可以互補行動資料格網之問題,我們將應用行動代理人技術與行動資料格網之系統建構上。

本篇論文主要的目標是在無線網路的環境下設計一個行動資料格網之架構,並在此架構下設計,根據節點頻寬、頻寬穩定度與剩餘量百分比提出一個適用於行動資料格網下的複本排程策略,達到高效率及可靠的資料傳輸服務,增加格網節點存活率,減少因斷線而造成的錯誤。本篇論文中,我們利用行動代理人技術,設計一個行動式資料格網架構,在此我們說明代理人中相關元件的動作流程。並且在這個架構下提供一個複本排程策略,命名為「Resource-Aware Replica Scheduling Algorithms」(RARS)。此複本排程演算法主要參考現存資料格網的共同配置(co-allocation)演算法的方式及行動裝置的剩餘能量,同時將資料傳遞分配到剩餘能量較高的行動裝置上,以提高行動裝置及系統的存活率。另外,也參考行動裝置的資料傳輸頻寬及資料傳輸的穩定度來分配資料傳遞的大小,以減少等待的時間,提昇傳遞的效率。

在本論文中我們首先在第二節中說明相關的技術及一些現存於有線網路下的複本排程演算法,這些演算法包含 Brute-Force Allocation、History based Co-Allocation 及 Dynamic Co-Allocation。在第三節中說明行動資料格網之架構及其元件,這些元件可以被建構在行動代理人中負責並且說明架構的動作流程。第四節中簡單說明我們所設計的行動資料格網節點的複本排程演算法。第五節我們分析我們的架構及演算法,並且與其他的演算法作一比較,分析其效能及優劣。最後一節我們給一結論。

## 二、背景及相關研究

### (一) 行動及無線格網

格網計算[25]由各種異質系統組成，目的是用來達成一個共同的目標 - 共享資源 [3, 9]。無線格網(wireless grid)[10]是一種新形態的資源共享網路，它能連接行動裝置和其他與無線格網相關的週邊設備，隨意地與分散式的資源共享。格網計算能讓設備連接到網際網路，並且覆蓋整個點對點式的網路與已存在的有線式格網計算動態地分享網路資源 [11]。無線格網延伸了這種分享的潛能，不管是移動的或者是固定位置的設備透過無線網路的連結，用戶端與設備便能進入一個動態的無線格網中，與經常性變動中的資源協同合作。行動格網是無線格網中最有趣的，由於加入了許多可利用、可分享的行動裝置到格網中，使得格網系統變得更複雜，但也造就了更強大的格網系統，當考慮到無數的無線設備時，格網的未來更具有潛力[12]。

McKnight[12]將無線格網基礎建設分為三個層次，分別為：實體層技術、網路基礎建設與中介軟體。作者提出行動格網的中介軟體研究主要議題，分別是資源描述、資源發現、協調與信賴建立。在本論文中我們主要集中在中介軟體議題上。無線格網中介軟體需要處理幾種挑戰，包括設備異質性，低頻寬和高延遲的網路連線等。設備的電源消耗也必須被重視。

資源描述是在格網計算中，任何的設備群組描述需要與可利用的資源的方法。例如，如果一組設備希望分享處理器週期，首先它必須能描述處理需求和能力 [12]。在資源描述方面，資源描述語言(RDL)[25]、標準化web服務描述語言(WSDL)<sup>1</sup>、頻寬描述從各種各樣的QoS標準可找到資源描述的協定。開放式格網服務架構(OGSA)<sup>2</sup>系統利用一種Web服務樣式目錄服務(Web service-style Index Service)收集每一個服務請求發佈的資源描述 [12]。Web服務社群也已經定義了以XML為基礎的WSDL、SOAP<sup>3</sup>、UDDI<sup>4</sup>和它的相關協定做為服務的標準。

為了有效地與行動裝置分享資源，協調系統必須提供資源存取控制的機制。協調系統允許一個設備利用另一種設備的資源，允許集中資源和調度資源。在一筆交易過程中的每一個資源甚至可能需要一種不同的類型的憑證建立 [12]。任何洽談過程必須能保證在過程期

間內交易伙伴的身分不能被改變，資源共享交易必須抵抗人為攻擊，以確保服務的品質與良好的安全性。

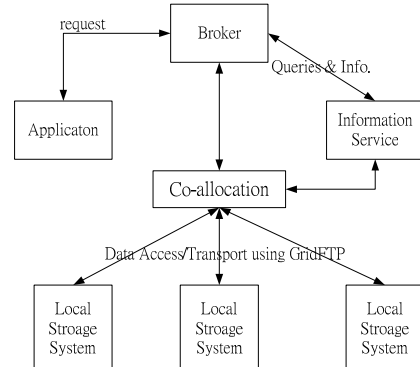
行動格網具有下列幾項特徵與挑戰[4, 11, 14]:設備的異質性、軟體系統的不同、低頻寬、高網路延遲、電池能量的消耗、迅速改變的網路拓撲、間斷式的網路連線、不固定的IP位址、斷線可能造成的錯誤。

### (二) 相關研究

資料格網能為大量資料密集的應用提供了分散式的服務及基礎建設，它被建立在分散式的儲存環境中，能存取、傳輸或修改巨量的資料集[1, 2]。資料格網的儲存系統在做資料、物件或檔案的管理上通常有一個複本管理系統，以達到傳輸及儲存管理上的效率[15]。格網的分散式儲存系統必須提供高效率且可靠的傳輸機制與複本的發現與管理機制。

Striped 資料傳輸協定 [16]使用多個資料串流來同時地存取檔案區塊，經由多個儲存節點中相同的複本同時地使用資料串流來存取不同區塊，透過這些節點分散了存取的負載，也改善了頻寬有限的情況 [8]。使用多條TCP串流來達到多個不同位置的來源端到單一接收端平行地資料傳輸，透過此種模式能聚集多個頻寬的能力，達到改善單一來源端有限頻寬的限制 [17, 18]。

為了更有效率地達成Striped 資料傳輸，Sudharshan Vazhkudai 在 [7]裡提出了共同分配的資料傳輸機制(Co-Allocated of data transfers)，圖二所示，應用程式向 Broker 描述它的需求，Broker 連結到 Information Services，找出合適的資料集，Information Services 將資料集相關的訊息傳送給交叉分配代理人(Co-Allocation Agent)根據設定的複本選擇策略與演算法，從格網多個複本節點平行地下載資料或資料集。



圖二：資源管理架構與 Co-Allocation 機制

Sudharshan Vazhkudai 也提到三個 Co-Allocation 複本選擇方法:Brute-Force、History-based 與 Dynamic Co-Allocation。Brute-Force 是此機制是將欲傳輸物件平均切割，並分配工作給有此物件的格網節點，如果

<sup>1</sup> <http://www.w3.org/TR/wsdl>

<sup>2</sup> <http://www.globus.org/ogsa/>

<sup>3</sup> <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

<sup>4</sup> <http://www.uddi.org/specification.html>

有  $n$  個節點，每一個節點位置所配分到的工作為檔案大小  $1/n$ 。

History-based Co-Allocation 是根據目前估測的頻寬來做為分配工作大小的依據，假設傳送的檔案大小為  $s$ ，存在此檔案的格網節點有  $n$  個位置，每一個節點  $i$  估測的頻寬為  $B_i$ ，則  $n$  個節點全部頻寬  $A$  為：

$$A = \sum_{i=1}^{i=n} B_i$$

最後每一個節點  $i$  分配到的工作大小為：

$$S_i = \frac{B_i}{A} \times S$$

Dynamic Co-Allocation 動態地交叉分配機制能根據頻寬的變化在每次分配時作調整。在 [7] 裡提到假如某一個節點的傳輸效率下降到一個臨界值，能將輸送的工作移至或重新分配到另一個可利用的資料流中，但是此種方式必須加入動態監聽資料流的能力這會造成額外的成本。此外，效率下降臨界值的準則是困難決定，因為這會根據網路或系統的情況而有所變化。Yang [6] 提出了另一個動態分配演算「The Recursive-Adjustment Co-Allocation」。此方式的目標為使每一個 Replica Server 的期望完成時間一致，這能有效減少傳輸快的 Replica Server 等傳輸較慢的 Replica Server。每一個 Replica Server 目前的頻寬來分配  $Section_i$  區段， $Section_i$  的預期完成時間為  $E(TI)$ ，然而實際的傳輸過程中，每一個 Replica Server 有可能超過預期的完成時間，所以在  $Section_{i+1}$  的分配中，其每一次分配的依據如下：

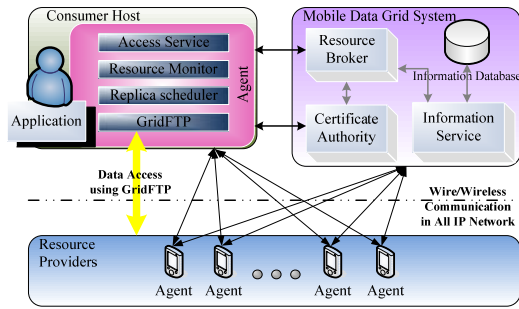
$$Server_i = \left( Section_j + \sum_{i=1}^n \frac{UnFinishSize_i}{BW_i} \right) * \frac{BW_i}{\sum_{i=1}^n BW_i} - UnFinsihSize_i$$

每一個 Replica Server 分配的 size 會扣除在  $Section_i$  中未完成的 size，以減少預期的差距。

### 三、行動資料格網

#### (一)行動資料格網架構

我們所提出的行動資料格網架構，如圖三所示，行動資料格網架構可分為三個參與者，分別為：資源需求者、資源提供者與資料格網系統，其中格網系統的部份包含了 Resource Broker、Certificate Authority 及 Information Service。



圖三：Architecture of The Mobile Data Grid

資源需求者與資源提供者都是行動格網中的一個節點，這些節點包含一個行動格網代理人(Mobile Grid Agent)，它負責與其他格網節點及格網系統溝通，每一個代理人包含了資源監控程式(Resource Monitor)、存取服務(Access Service, AS)、複本排程模組(Replica Scheduling model)、GridFTP 傳輸機制，及其他相關的服務，透過格網代理人每一個節點能向格網系統取得信賴憑證或透過 GridFTP 與其他格網節點傳輸資料。

#### Resource Broker

它用來做資源收集與協調的工作，當一個資源消費者需要一項資源時，必須先聯繫 Resource Broker，說明需求的資源型態，Resource Broker 再找出與需求符合的資源。

#### Certificate Authority(CA)

Certificate Authority 負責發行信賴憑證的服務，例如 x.509 憑證標準下發行的憑證或採用 Kerberos 等安全認證標準。由於格網應用在跨組織、分散式環境下，資源將會被許多不同的組織、不同地點存取，格網需要確保資料的完整性與保密性，在分散式環境下訊息傳遞採用標準加解密過程是必要的。

#### Information Service

Information Service 負責記錄資源的各項訊息與提供目錄查詢服務。當 Resource Broker 接收資源提供者的資訊時，它必須把資源的相關訊息記錄在 Information Service，以供日後查詢。資訊更新使得格網系統的訊息能保持在最新的狀態，準確的資源訊息提供其他需求者獲得最佳的資源選擇與分配，訊息更新依據不同的資源型態、不同的應用需求，會有不同的更新機制。訊息更新的方式有下列幾種：

1. 格網節點定時地回報最新狀態。
2. 當資源狀態有變化時回報最新資訊。
3. 當需求者查詢時，由 Information Service 要求特定的格網節點回報最新的資訊。

#### 資源監控程式(Resource Monitor)

資源監控程式(Resource Monitor)能監控資源的最新狀態，例如設備可利用的計算資源、設備的儲存資源、設備的網路頻寬、複本

的版本資訊等。監控程式能要求行動設備的電源管理程式回報目前剩餘電量百分比；要求系統工作管理回報 CPU 使用率；透過頻寬監控取得目前可利用的頻寬等。

### 存取服務 (Access Service, AS)

存取服務 (AS) 負責對格網節點的資源做存取的控制，防止非法存取格網的資源。要使用行動資料格網的資源，必須向 Mobile Data Grid System 的 CA 取得對格網節點的存取授權書，才能使用此節點的資源。資源使用者從 Information Service 取得複本列表，複本列表包含每一個複本資訊，如複本節點的位址、資源的識別碼或名稱，此外 CA 會給與另一項重要的資訊，每一個複本節點的「存取授權書」。存取授權書為一組透過公開鑰匙加密的訊息，訊息必須包含下列資訊：授權帳號、存取權限、存取資源、存取期限。當使用者請求資源時，CA 會給予資源消費者每一個資源提供者的存取授權書，資源消費者要求資源提供者提供檔案時，會先將存取授權書傳遞給資源提供者，資源提供者收到存取授權書後，AS 驗證此授權書的正確性及授權期限，並記錄在 AS 的存取目錄上。當授權期限到期，再從存取目錄中移除。

### 資源排程模組 (Resource Scheduling module)

資源排程模組 (Resource Scheduling module) 可依據取得的資源列表，選擇最適合的資源來提供服務。不同的應用有不同的選擇依據，例如，依據應用選擇合適的計算資源。依據資源利用率設備選擇合適的儲存地區。依據設備電池的電力選擇適合的複本資源等。

### 格網檔案傳輸協定 (GridFTP)

GridFTP 為資料格網的檔案傳輸協定，可利用 Striped 資料傳輸平行地從多個不同位置的來源端傳輸資料複本，配合複本選擇演算法，提供高效率且可靠的傳輸機制。

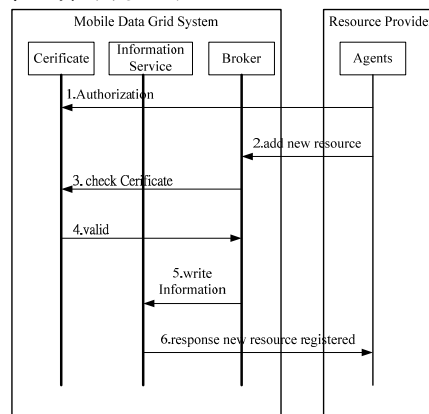
## (二) 新資源的加入

新資源的加入行動格網中是相對頻繁的，當使用者同意設備加入格網中或將設備資源列入格網，格網系統就必須註冊此項資源，並取得資源的訊息，以供日後索引查詢。圖四說明新資源加入的流程：

1. 資源提供者登入，並取得憑證。
2. 當提供者提供新資源到格網，資源提供者向 Resource Broker 描述設備名稱、網路頻寬、電池續航力、提供的資源型態、資源狀態、存取限制等資訊。
3. Resource Broker 要求 CA 檢查資源提供者的身分憑證
4. CA 回報憑證的合法性
5. Resource Broker 註冊新資源 Information Service，Information Service 給予資源

的識別碼，並將這些記錄儲存到 Database。

6. 最後，Resource Broker 回報完成作業程序給資源提供者



圖四：新資源加入的過程

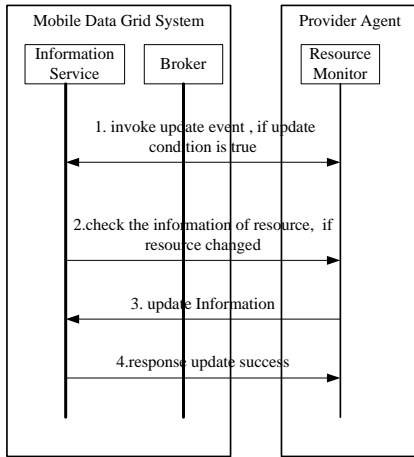
## (三) 資源訊息的更新

資源訊息的更新使得格網系統的訊息能保持在最近的狀態，準確的資源訊息提供其他需求者獲得最佳的資源選擇與分配，訊息更新依據不同的資源型態、不同的應用需求，會有不同的更新機制。更新的方式有下列幾種：

- 格網節點週期地回報最新狀態。週期地回報能讓資訊保持在最近的狀態，例如：即時性的應用中使用週期地回報頻寬狀態。
- 當資源狀態有變化時回報最新資訊。例如，檔案的版本資訊，當一有版本更新，則將版本資訊回報給 Information System。
- 當需求者查詢時，由 Information Service 要求特定的格網節點回報最新資訊。例如：Information Service 要求特定的格網節點回報目前 CPU 使用率。

圖五說明訊息更新流程，訊息依據不同的更新條件，其更新過程如下：

1. 當更新條件觸發時，由 Information Service 或行動代理人提出更新。
2. 檢查資源的相關資訊是否有變更，如果有則執行更新。
3. 執行更新上傳最近訊息，更新的訊息必須包含資源識別碼、資源狀態、更新的時間戳。
4. 最後，系統回報更新完成。



圖五：資源訊息更新流程

#### (四)資源的請求

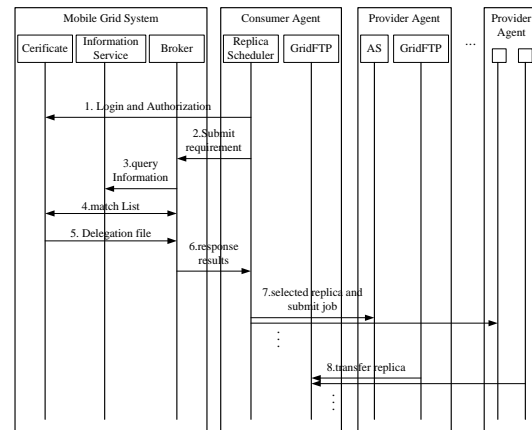
資源的請求為我們最主要的情節，當使用者請求使用格網資源時，必須將資源需求描述給 Resource Broker，再由 Resource Broker 回覆合適的資源列表。圖六為使用者請求資料複本的過程：

1. 首先使用者登入，並向 CA 取得認證。
2. 當一個應用程式需要存取格網資料時，應用程式必須描述需求，如資源的型態、資源狀態等訊息。透過格網代理人傳遞給 Resource Broker。
3. Resource Broker 向目錄資訊服務 (Information Service) 查詢合適的資源。
4. 將合適的資源列表回覆給 Resource Broker 與 CA。
5. CA 取得複本列表，製作每一個複本的存取授權書，回覆 Resource Broker。
6. Resource Broker 取得複本列表與每一個複本的存取授權書，一併傳回給資源消費者 (consumer)。
7. 最後 consumer 端的格網代理人 (Agent)，透過複本排程器來選取複本列表中的資源提供者，並連線到選取的資源提供者，consumer 端格網代理人將存取權限書  $E_{u_i}(A_i)$  傳遞給資源提供者  $g_i$ 。
8. 資源擁有者用私鑰解開權限資訊  $A_i$ ，將存取權限加入存取目錄並開啟權限，consumer 端的格網代理人 (Agent) 使用 GridFTP 傳輸協定下載所需的檔案資源。

#### 四、動態複本排程演算法

傳統的複本選擇，如：Brute-Force Co-Allocation、History-based Co-Allocation、Dynamic Co-Allocation 等方法會將工作分配給所有可利用的複本節點，以提高存取速度，這些方法的確改善了傳輸的

效能。然而，在無線格網的環境中，設備會經常性地加入或離開，每一個時間點可利用的複本節點數不會相同，動態的移動節點及間斷性的連線更使得頻寬與連線相對地不穩定。此外，在無線裝置上亦必須考慮到電力的存量，盡量使用電池能量較多的設備將有效地提高無線格網的可用性，無線格網提供了大量不同能力的設備，我們不必用盡所有可用的節點，我們可以選擇較穩定、更適合的設備來為我們提供服務。根據上述的因素，我們提出的一個複本選擇演算法「Resource-Aware Replica Scheduling Algorithms」(RARS)，並且使用節點資源描述檔 (peer profile) 來描述每一個格網節點的能力。RARS 加入傳統複本排程策略沒有考慮到電力的存量的因素，透過減少使用電量少的用戶端設備，使得整體網路存活率提高，達到無線資料格網的可用性。改善無線通訊因斷線而無法完成工作的情况。



圖六：複本請求的過程

#### (一)節點資源描述

每一個行動裝置的能力都與其他的不同，3GPP 定義了行動裝置的 CC/PP [21] 用以描述其靜態的能力，不過行動裝置的動態能力，像是剩餘的能量、連線的頻寬及裝置的穩定性等，這些都對於決定到那一個裝置存取資料是很重要的。因此下面我們定義了一些有關於行動節點的動態描述 (Dynamic Profile: DP)，來表示目前行動裝置的資源，RARS 可以利用此 DP 來動態的決定由那些複本存取資料。 $P_i$  表示節點  $i$  的 DP，包含四個資訊。

- 頻寬 ( $\beta$ ):  $\beta_i$  表示第  $i$  個節點目前的傳輸頻寬。
- 剩餘能量 ( $E$ ):  $E_i$  表示節點  $i$  的剩餘能量。
- 連線穩定度 ( $M$ ):  $M_i$  表示第  $i$  節點的連線穩定度。
- 較低的能量邊際 ( $E_{Thr}$ ):  $E_{Thr}^i$  表示節點  $i$  允許其他節點存取資料的最低能量，如果  $E_i < E_{Thr}^i$ ，則不允許其他的節點存取

這個節點的資料。

$\beta_i$  及  $E_i$  都可以在傳輸資料的同時附帶由資料來源節點傳回或由 Network Weather Service (NWS) [22]來回報每一個節點的相關資源。委託端可以藉此來判斷是否繼續存取這個節點的資料及下一個請求的資料區塊大小。 $M_i$  則是由委託端來監督使一個來源的穩定性，如果每一次存取資料時可在預期時間內傳回，則  $M_i$  加 0.1，否則減 0.1。

## (二) 配置函數

描述配置函數前先定義三個參數：

- **封包大小  $\theta$** ，一個封包大小是固定的，而每次分配給複本要傳輸的資料都是數個  $\theta$ 。
- **配置的傳輸時間  $T_p$** ，某一回合的資料傳輸時間，這個觀念同於 dynamic co-allocation 方法，我們期待每一回合中所有複本都有相同的資料傳輸時間，理由是委託端有固定的時間檢查無線網路的連線，這個方法可以早期偵測行動裝置是否由於通訊或連線的不穩定而間歇性斷線。

$$T_p \geq \left\lceil \frac{\theta}{\text{Min}(\beta_0, \dots, \beta_n)} \right\rceil \quad (1)$$

- **期待的延遲  $d_{i,k}$** ，第  $i$  個節的第  $k$  次請求期待的時間延遲，用來檢查一個連線是否超出時間，如果超出時間則未傳完的區塊都會重新分配給其他節點，我們使用 TCP[23]的方式來估計  $d_{i,k}$  的值，如下：

$$d_{i,k} = \alpha \times d_{i,k-1} + (1 - \alpha) \times |T_{i,k-1} - \rho_{i,k}|$$

$T_{i,k-1}$ : 上一次工作完戶時間

$\rho_{i,k}$ : 估計的完成時間

$\alpha$ : 平滑因子，一般為  $\alpha=7/8$

一開始  $d_{i,0}$  設成  $T_p$ ，timeout 可為如下：

$$\text{Timeout} = \rho_{i,k} + 4 \times d_{i,k}$$

如果一個複本的服務時間超過 timeout 時間，則複本排程器會感知這個連線是一個間歇性連線或斷線，則這個複本的  $M$  將遞減，否則  $M$  將遞增。

接著介紹配置函數，此函數用以計算由複本傳遞資料的大小。配置的大小可以根據裝置的資源來計算，像是頻寬及穩定度。而剩餘能量則用來決定存取的優先權。節點  $i$  的第  $k$  個配置可以如下：

$$\tau_{i,k} = \left\lfloor \frac{T_p \times \beta_{i-1} \times M_i}{\theta} \right\rfloor, \quad M_i = 1 \pm \delta, \quad M_i \geq 0 \quad (2)$$

$M_i$  是用來微調存取的大小，根據委託端及格網節點間連線的穩定度來決定。較小的  $M_i$  表示較高的間歇性連線機率，則順利存取的機率較低，需等待傳遞完成的機率較高，因此不要給予太多的區塊傳輸。 $\delta$  邊界參數，其值為 0.5。

$\rho_{i,k}$  是期待完成時間定義如下：

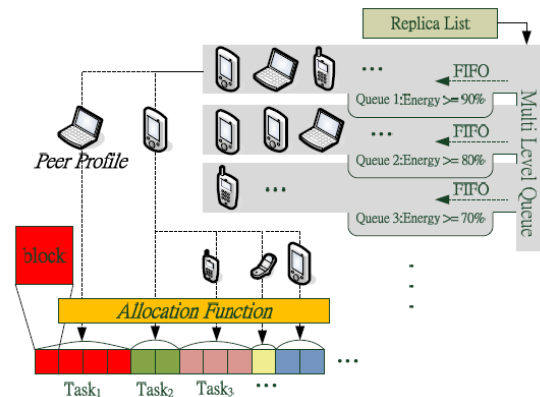
$$\rho_{i,k} = \tau_{i,k} \times \theta \div \beta_{i,k-1} \quad (3)$$

RARS 會檢查如果存取超出 timeout，如果超出則會將這些區塊分配給其他節點，以改善間歇性連線問題。

## (三) 複本選擇與排程

RARS 也是一種共同配置(co-allocation)的演算法。這個演算法除了有效率的傳輸資料外，亦可增強系統的可用性及可靠度。描述 RARS 之前先定義在複本排程器中的一個資料結構，稱為多層能量佇列(Multiple energy level Queue (MQ))，它是一個多層的佇列，被放置在每一個佇列中的節點代表有相同層次的剩餘能量，這個 MQ 共分成八個，分別是 100%~91%、90%~81%、...、及 30%~21% 剩餘能量的節點，被放置在這些佇列的表示有委託端所需要的複本，且該節點的剩餘能量分別是該佇列所表示的。

當委託端由 Information service 取得所要資料的所有複本節點及所有的 DP 時，委託端可將這些複本節點放到相對應的能量佇列中，用來決定要取得資料的優先順序，如圖七所示。複本排程器會優先選擇能量較高的複本節點來存取複本的資料，接著利用配置函數來配置複本所要傳輸的資料區塊大小，如果最高剩餘能量佇列中的複本節點都分配完，再選擇下一個層級的佇列，一直到所有的複本都分配完或所分配的傳輸頻寬已經超過委託端所能接收的頻寬。



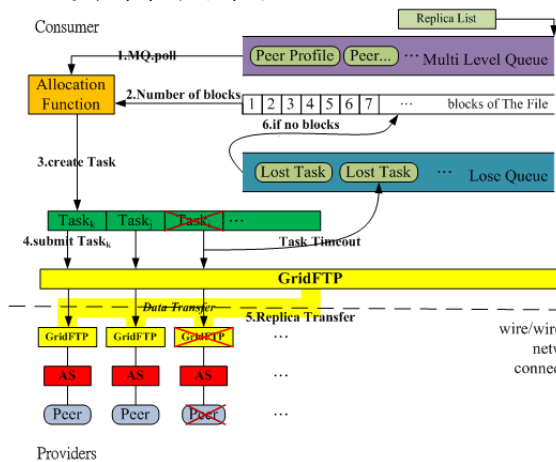
圖七：RARS 中複本的選擇

當複本所配置的資料傳輸完畢後，委託端會根據資料傳遞後所回報的剩餘能量值來決

定要將複本節點放置到那一個對應的MQ佇列中，再配置剩餘的複本資料區塊。

行動環境中經常的間歇性連線會造成資料的遺失及降低系統的效能，因此改善系統的可靠度是另一個重要的議題。RARS 使用另一個工作遺失佇列 (Job Lost Queue, 簡稱 LQ) 來追蹤間歇性連線的問題。LQ 是負責保存複本的 profile 及未傳輸完的區塊，當複本排程器偵測到 timeout 的事件時，它記錄已配置且未完成的區塊到 LQ 中，如圖八所示。

在 RARS 中，在 LQ 中的工作有最高的優先權，它會先被配置到其他的節點，直到這些未完成的區塊被分配到 MQ 中的節點後，才會分配其他的節點，這個機制可以減少當斷線或間歇性連線時等待的時間。



圖八：當傳輸工作未完成的劇情

下列二個演算法是RARS主要的演算法及所使用的配置函數。

#### Algorithm: RARS main function

##### Input:

FileName desiredFile;  
BlockSize  $\theta$ ;

##### Output:

None;

1.  $accessedBW \leftarrow 0$ ,  $MQ[0 \dots m-1] \leftarrow empty$ ,  $LQ \leftarrow empty$ ;
2.  $ReplicaList \ r \leftarrow getReplicas(desiredFile)$ ;
3.  $While(r.next())$
4.  $MQ.set(r.getProfile().remainingEnergy)$ ;
5.  $End\ while$
6.  $minBW \leftarrow getMinBW()$ ;
7.  $Tp = \theta / minBW$ ;
8.  $While(!JobFinished \ \& \ accessedBW \leq totalBW)$
9.  $While(!LQ.empty())$
10.  $Blocks \ b \leftarrow LQ.getLostBlock()$ ;
11.  $AddBlocks(b)$ ;
12.  $End\ while$
13.  $For(i=0; i \leq m-1; i++)$
14.  $Iff(accessedBW \leq totalBW$

$\& \ !MQ[i].empty())$

15.  $Profile \ p \leftarrow MQ[i].getNextReplica()$ ;
16.  $AllocateTask(p, Tp, \theta)$ ;
17.  $accessedBW += p.Bandwidth$ ;
18.  $End\ if$
19.  $End\ for$
20.  $End\ while$

#### Algorithm: Allocation function

##### Input:

Profile  $p$ ;  
Time  $tp$ ;  
Size  $\theta$ ;

##### Output:

None;

1.  $bw \leftarrow p.getBW$ ,  $E \leftarrow p.getRemainingEnergy$ ,  $M \leftarrow p.getStability$ ;
2.  $allocatedBlocks \leftarrow computeSize(bw, E, M, tp, \theta)$ ;
3.  $numberOfBlock \leftarrow allocatedBlocks.max - allocatedBlocks.min + 1$ ;
4.  $InvokeAccessing(p, numberOfBlock, allocatedBlock.max, allocated.min)$ ;

## 五、模擬與效能分析

行動資料格網的複本傳送主要考量二項成本：傳輸速度與無線裝置的用電量；在真實的無線網路環境，頻寬隨著使用者的行為與環境的變動而有所改變，包含不同的通訊技術、使用者的移動模式、移動的速度、使用習慣、使用頻率、與基地台或 AP 的距離、建築物的阻隔等，都會影響頻寬的變化。由於各種複雜的環境條件都可影響頻寬變化且影響頻寬變化的因素多而複雜，故我們僅使用非一般性的假設，我們假設每一個移動裝置的頻寬變化都會有不同的 Gaussian 分佈變化，來表示動態環境的頻寬變化。

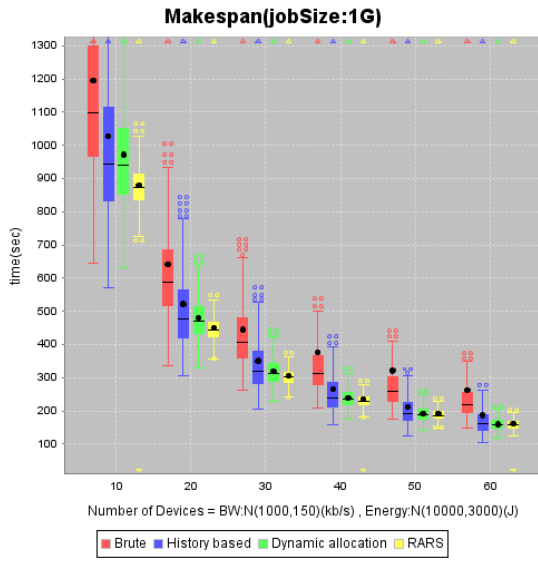
我們模擬並且比較 RARS 複本排程演算法及另外三種目前被提出的複本傳送方法如下：

1. Brute-Force Co-Allocation
2. History-based Co-Allocation
3. Dynamic-Adjustment Co-Allocation

效能分析的模擬中，我們使用上述各種的複本排程方法測量在頻寬變化上之最短完成工作時間。分別模擬 10、20、30、40、50、60 台無線裝置，並且假設在各種網路情況下的頻寬狀態下，讓每一個設備有不同的標準差的 Gaussian 分佈及不同設備的電源。

我們將 Dynamic Co-Allocation 演算法的參數  $\alpha$  值設為 0.5， $\alpha$  值為每次分配檔案的百分比， $block\ size$  設為 1 kb。最後，各別設定傳送的檔案大小為 1GB。我們讓不同類型的網路頻寬、不同類型的設備電源，及各種複

本排程演算法來傳送相同大小的檔案，以評估它的傳輸速度效能。



圖九：完成工作時間

圖九中我們可以明顯地觀察出平均的傳輸時間 RARS 演算法效能略快於 Dynamic Co-Allocation 演算法，其次是 History based Co-Allocation，最差的為 Brute-Force Allocation。我們可以看到在節點數為 10 時，History based 平均花費時間約為 6000 秒，而 Dynamic 演算法，平均花費時間約為 5800 秒，RARS 平均花費時間約為 5400 秒。二者相較之下，RARS 平均花費時間約減少 0.06%。在實際情況下，無線網路的設備會經常性地加入或離開，每一個時間點可利用的複本節點數不會相同，這對於 Dynamic Co-Allocation 與 History based Co-Allocation 是一項缺點，因為 Dynamic Co-Allocation 與 History based Co-Allocation 必須統計所有複本節點的頻寬值，但在行動格網下所有節點頻寬值之總和是不容易正確地統計出來的，為了能與其它演算法比較，模擬的方式使我們容易計算頻寬數據之總和。然而 RARS 演算法在此點上取得優勢，因為它不需統計頻寬總和，即使已排入佇列的複本節點退出網路，在我們從佇列中取出它時，確認它退出網路，將它排除即可。

在可用度分析中，我們使用各種的複本排程方法測量網路節點存活率。模擬 100 個格網節點，每一個節點平均頻寬在 300(kb/s)，不同的頻寬變化率，分別傳送三種不同檔案大小的工作。每一個複本排程演算法所使用的設備電量相同。我們比較在不同網路頻寬變化率、不同的檔案大小，及各種複本排程演算法評估最後的網路節點存活率。

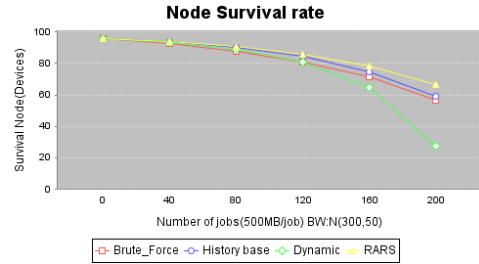


圖 5.4：網路節點的存活率 job size = 500MB

我們假設在節點數目固定不變下，當整個網路執行愈多工作時，節點數目會慢慢減少。我們可以明顯地觀察出節點存活曲線下降的弧度 Dynamic Co-Allocation 最大，其次為 Brute-Force Allocation，其次為 History based Co-Allocation，RARS 演算法節點存活曲線維持最長久，我們亦可觀察出當頻寬變化率愈高時，Dynamic Co-Allocation 的節點存活曲線下降愈快；這是因為當傳輸的檔案較小時，每次傳送小檔案 Dynamic Co-Allocation 過於利用高頻寬的節點，卻忽視了嚴重消耗設備電量的結果，使得節點存活曲線比其他演算法更提早下降。換言之執行 Job 數目愈多，代表格網系統運作愈久，以圖十來看在 200 個 Job 時，RARS 節點數約還有 65 個，而 Dynamic 節點數只剩下約 25 個，其差距整體節點數的 60%。

表一為各項演算法之比較表。依據分配回合數來區分，Brute\_Force 與 History\_Base 演算法為一次分配全部工作，Brute\_Force 演算法以節點數目等量分配，而 History\_Base 演算法以一開始獲得的頻寬為依據來分配工作，在傳輸期間頻寬變化與一開始獲得的頻寬數據差異很大，則工作傳輸時間可能與預估的完成時間差距很大；而 Dynamic 與 RARS 演算法以區段或週期性做 N 次的調整工作分配，因此能有較佳的傳輸效率。

表一：演算法比較表

	Brute-Force	History base	Dynamic	RARS
Number of Rounds	one	one	N	N
Time of a Round	variable	variable	variable	fixed
Allocation metrics	Number of server	History BW	BW	BW, E%, M
Tolerant Node disconnection	no	no	no	yes
Makespan	worst	third	second	best



Node Survival Rate	third	second	worst	best
--------------------	-------	--------	-------	------

以每一回合的分配的工作量探討之，Brute\_Force 演算法以等量分配方式並不符合實際頻寬，造成工作分配不均衡；History\_Base 演算法以分配時得知之頻寬做一次全部分配，但在傳輸過程中，頻寬可能會有所變化，一樣地不符合實際頻寬變化而造成工作分配不均衡；Dynamic 演算法使用遞迴方式決定每一回合分配的區塊大小，開始分配的區塊最大，之後慢慢遞減，它的缺點是若一開始所分配的區塊大小太大，且剛好節點實際頻寬與預估值差異很大時，一樣容易造成工作分配不均衡的情況；RARS 演算法以時間為分配的單位，對每一個節點而言，每一次分配的工作量都在一段很小的時間內完成，實際頻寬與預估值差異並不會太大，即使頻寬值差異很大，在工作逾期時會重新分配此工作，並不容易造成分配不均衡的情況。

以工作量分配考慮的因素來比較之，Brute\_Force 演算法的分配依據為節點數目，History\_Base 演算法的分配依據為過去傳輸所計算出的頻寬值來分配，Dynamic 演算法每一回合重新計算頻寬值來動態調整分配量，而 RARS 考慮行動資料格網節點的特性，故工作量分配的依據有每一次分配時的節點頻寬值、電池能量百分比、節點的穩定度，因此更適應於無線通訊下的資料格網。

在離線容錯方面，Brute\_Force、History\_Base 與 Dynamic 演算法主要設計給傳統的有線網路，然而在無線通訊下的資料格網，行動節點離線是經常性發生地，故我們加入了工作逾期機制來解決因離線所造成工作無法完成的情況。

最後在模擬的結果方面，從表一的比較表中，不管是從完成工作的效率或是節點存活率的比較，RARS 演算法都呈現很好的結果。另一項值得探討，10 個節點傳輸資料的模擬時，RARS 演算法平均完工時間都快於 Dynamic 演算法，但是在節點愈多時，部份模擬結果卻顯示 Dynamic 演算法的平均完工時間略快於 RARS 演算法。呈現這樣的結果是因為在節點數目少時，RARS 演算法與 Dynamic 演算法都會盡量地利用複本節點，而在節點數目多的情況下，RARS 演算法使用電池能量較佳的節點，部份電池能量較低的節點可能分配不到工作，而 Dynamic 演算法將每一回合的工作分配給所有複本節點，所以在模擬的結果上 Dynamic 演算法的平均完工時間會出現略快於 RARS 演算法的結果。這並非為 Dynamic 演算法的優勢，雖然在完工效能上有機會加快完成，但它使用了這些電池能量較低的節點，相對地這些節點

無法完成工作的機率較高，若無法完成工作，既是浪費了這些節點的資源，而且還會降低行動用戶對資料格網的使用率。

## 六、結論

我們提出一個行動資料格網的架構，並且在這個架構下提供一個資料複本傳輸的排程演算法，這個架構透過用戶端代理器(Agent)能與格網系統溝通取得複本資訊。經過模擬，並比較其他傳統的複本排程演算法，我們所提出的複本傳輸方法能達到高效率的資料傳送，且透過電量等級的區分能減少使用到電量較少的用戶端設備，使得整體網路存活率提高，達到行動資料格網的可用性。我們所提出的方法亦加入了「工作完成期限」，當複本端超過工作完成期限，此項工作會重新分配給另一個複本端，能改善無線設備因斷線而造成工作無法完成的情況，提高了行動資料格網的可靠度。

## 參考文獻

- [1] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury and S. Tuecke, "The data grid: Towards an architecture for distributed management and analysis of large scientific datasets," *Journal of Network and Computer Applications*, 2000, pp. 187-200.
- [2] W. Hoschek, F. J. Jaen-Martinez, A. Samar, H. Stockinger and K. Stockinger, "Data management in an international data Grid project," *In Proceedings of the 1st IEEE/ACM International Workshop on Grid Computing (GRID'00)*, 2000, pp. 17-20.
- [3] T. Phan, L. Huang and C. Dulan, "Challenge: Integrating wireless devices into the computational grid," *International Conference on Mobile Computing and Networking*, Aug. 2002, pp. 23-28.
- [4] Agarwal, D. Norman and A. Gupta, "Wireless grids: approaches, architectures and technical challenges," *MIT Sloan School of Management*, Jan. 2004.
- [5] D. Chu and M. Humphrey, "Mobile OGS.NET: grid computing on mobile devices," *IEEE/ACM International Workshop on Grid Computing*, Nov. 2004., pp.182-191.
- [6] Chao-Tung Yang, I-Hsien Yang, Chun-Hsiang Chen, and Shih-Yu Wang, "Improvements on Dynamic adjustment mechanism in co-allocation data grid environments," *J. of Supercomputing*, (2007) 40: 269-280.
- [7] Vazhkuda and Sudharshan, "Enabling the co-allocation of grid data transfers,"

- Fourth International Workshop on Grid Computing*, Nov. 2003, pp. 44-51.
- [8] S. Vazhkudai, S. Tuecke, and I. Foster, "Replica Selection in the Globus Data Grid." *Proc. of the 1st International Symposium on Cluster Computing and the Grid (CCGRID 2001)*, May 2001, pp. 106-113.
- [9] C. Thompson. "Agents, grids and middleware," *IEEE Internet Computing*, Vol.8 No.5 , Sept. 2004, pp. 97-99.
- [10] Howison and L.W. McKnight, "Towards a Sharing Protocol for Wireless Grids," *Proc. Int'l Conf. Computer Comm.and Control Technologies, Int'l Inst. of Informatics and Systemics, Orlando, Fla., vol. 000648*, 2003.
- [11] Gaynor and L.W. McKnight, "Wireless Grid Issues," *8th Global Grid Forum (GGF8)*, [Online] 2003. <http://www.ece.tufts.edu/~hchang/Projects2003/draft-ggf-lwmcknight-wgissues-0.pdf>.
- [12] L. McKnight, J. Howison and S. Bradner, "Wireless Grids--Distributed Resource Sharing by Mobile, Nomadic, and Fixed Devices," *IEEE INTERNET COMPUTING*, Jul. 2004, pp. 24-31.
- [13] G. Xing, C. Lu, R. Pless and Q. Huang, "On greedy geographic routing algorithms in sensing covered," *5th ACM international symposium on Mobile ad hoc networking and computing*, May 2004, pp. 31-42.
- [14] Huang Chang-Qin, Zhu Zhi-Ting, Wu Yong-He and Xiao Zheng-Hong, "Power-Aware Hierarchical Scheduling with Respect to Resource Intermittence in Wireless Grids," *Machine Learning and Cybernetics, 2006 International Conference on*, Aug. 2006, pp. 693-698.
- [15] Srikumar Venugopal, Rajkumar Buyya and Kotagiri Ramamohanarao, "A taxonomy of Data Grids for distributed data sharing, management, and processing," *ACM Computing Surveys (CSUR)*, June. 2006.
- [16] W. Allcock, J. Bresnahan, R. Kettimuthu and M. Link, "The Globus Striped GridFTP Framework and Server," *Proceedings of the ACM/IEEE SC 2005 Conference*, Nov. 2005, pp. 54-54.
- [17] T. Hacker, B. Athey and B. Noble, "The end-to-end performance effects of parallel tcp sockets on a lossy wide-area network," *16th IEEECS /ACM International Parallel and Distributed Processing Symposium*, 2002, pp. 434-443.
- [18] L. Qiu, Y. Zhang and S. Keshav, "On Individual and Aggregate TCP Performance," *7th International Conference on Network Protocols*, 1999, pp. 203-212.
- [19] Chao-Tung Yang, Chun-Jen Chen and Ching-Hsien Hsu, "A Peer-to-Peer Resource Sharing System with Data Grid Technology for Mobile Devices," *International Conference on Multimedia and Ubiquitous Engineering (MUE '07)*, Apr. 2007, pp. 723-728.
- [20] Xiaodong Yan, Huimin Xu, Yajing Xu and Lianxi Liu, "A data replica replacement algorithm based on value model in mobile grid environments," *2nd International Conference on Mobile Technology, Applications and Systems*, Nov. 2005, pp. 15-17.
- [21] Graham Klyne, Franklin Reynolds, Chris Woodrow, Hidetaka Ohto, Johan Hjelm, Mark H. Butler, and Luu Tran, "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0." [Online] 2004. <http://www.w3.org/TR/CCPP-struct-vocab/>
- [22] R. Wolski, N. Spring, and J. Hayes, "The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing," *Future Generation Computer Systems*, 15(5-6):757-768, 1999.
- [23] Andrew S. Tanenbaum, "Computer Networks", 4th edition, Pearson Education International.
- [24] Danny B. Lange, Mitsuru Oshima, "Seven good reasons for mobile agents," *Communications of the ACM*, Vol. 42 No. 3, 1999, pp.88-89.
- [25] Klaus Krauter, Rajkumar Buyya, Muthucumar Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing," *Software: Practice and Experience*, Vol. 32, No. 2 , 2001, pp. 135 - 164