在資料網格環境下
以服務導向架構提升資料儲存的效能

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

呂彗君
朝陽科技大學
資訊與通訊系 研究生
s9730614@cyut.edu.tw

嚴國慶*
朝陽科技大學
企業管理系 教授(聯絡人)
kqyan@cyut.edu.tw

摘要

網格(Grid)是一種計算跟資源集合的環境，能夠充份的利用各種可供計算的資源，將其轉換成隨處可得、標準的、具經濟效益的計算能力。在目前的網際網路，許多的機構或是企業，都需要一個存取大量資料的設備，但是這些設備的建置往往需要龐大的成本。在資料網格(Data Grid)的環境下，擁有眾多的可用資源，這些資源是異質性的或是同質性的，利用這些資源，可以達到資源有效的儲存，並同時降低在管理資料檔案時所需花費的成本。因此，本研究提出 SOS(Service Oriented Storage)架構，除了可以提供有效的檔案儲存外，並且以最有效率的方式找出合適的儲存節點(Storage Node)，讓使用者在進行檔案儲存時，能夠有效的達到滿意的儲存結果。

關鍵詞：資料網格、資源分享、分散式資料處理、服務導向

Abstract

In recent years, network bandwidth and quality has been drastically improved, even much faster than the enhancement of computer performance. The various communication and computing tasks in the fields such as telecommunication, multimedia, information technology, and construction simulation, can be integrated and applied in a distributed computing environment in nowadays. However, data grid is to utilize the storage resources on the network to facilitate the resource sharing. In this study, a Service Oriented Storage (SOS) framework is proposed to support an efficient data store.

Keywords: Data Grid, Resource Sharing, Distributed Computing**,** Service Oriented

1. 前言

網格(Grid)是一種計算跟資源集合的環境，能夠充份利用各種可供計算的資源，將其轉換成隨處可得、標準的、具經濟效益的計算能力，這種計算能力稱之為網格運算(Grid computing)。在傳統的資源管理系統中，大多數只能做到集中式的管理，可是在網格的環境下，因為設備的佈建為分散且不規則式，因此不適合集中式的管理方式。但是在網格的環境中，可以利用同質性或異質性的資源設備，來處理各種切割成片段的運算問題，因此它具有兩種優勢：(1)資料處理的能力很強，(2)能夠充分的利用網路上的閒置資源[1]。網格的資源可以位在不同的地理位置上，其含蓋的範圍可從地方性到國際性。除了資源計算之外，網格計算還提供了資料存取、分享、與管理等服務。實現計算資源、資料資源、資訊資源、知識資源、專家資源、與儲存資源的全面共用[2]。目前在科學界、工程學界以及學術界等都有網格的相關應用。從應用的觀點來看，可以將網格運算分成計算網格(Computing Grid)、資料網格(Data Grid)和服務網格(Server Grid)。

由於網際網路的普及，許多機構或是企業組織，都需要一個存取大量資料的設備，但這些設備的建置往往需要龐大的成本。而在 Data Grid 的環境下，擁有眾多的可用資源，這些資源可以是位於不同地理位置的異質性或是同質性的資源。每一個資源都會提供容量大小不一的可用空間，在進行檔案儲存時，除了檔案的正本會儲存在資源中某一節點之外，還會有稱之為副本的備份檔案，同時儲存在另一資源中的節點。為提供各機構或是企業組織所需的儲存資源，減少這些機構在建置資料儲存設備的成本，也提高資料儲存的穩定性及安全性，在本研究中將假設這些儲存資源是自主性的提供其可用空間，其可信度與忠誠度極高、安

全性很好、變動性很低，亦即每個資源的穩定性都可被接受。換言之，本研究將建置在 Data Grid 的網路環境之上，圖 1 所示為 Data Grid 的環境。

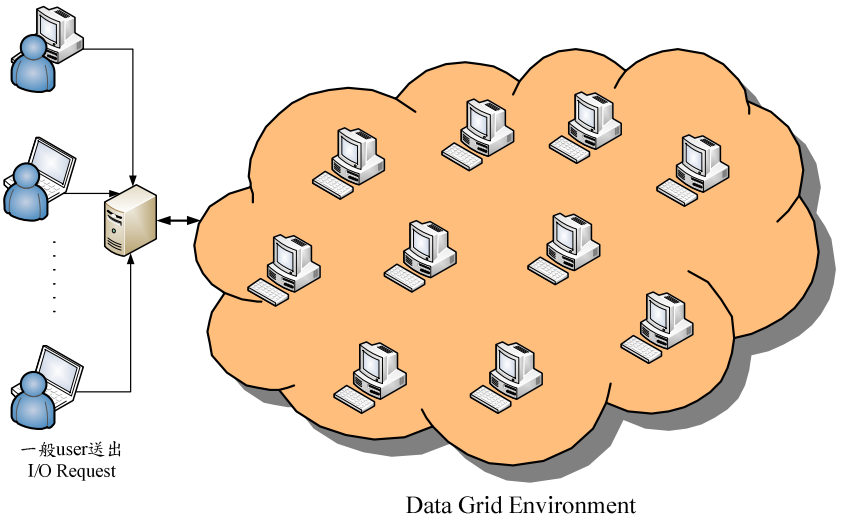

圖 1、Data Grid 環境

為提供各機構或企業組織所需的儲存資源，本研究提出 SOS(Service Oriented Storage)架構，其目的是為了要有效的進行檔案的儲存，並且以最有效率的方式找出合適的儲存節點(Storage Node)，讓使用者在進行檔案儲存時，能夠有效的達到讓使用者滿意的儲存結果。本文在第二節將針對目前網格的概況進行介紹。第三節則說明本研究所提出的 SOS 系統架構。第四部份則是總結及未來的工作。

2. 文獻探討

在網格計算的發展中，Globus[3]是目前在 Grid 社群中軟體發展的標準之一，但 Globus 並非是統一的標準。Globus 在網格的工具上，發展出可供使用者更方便建立分散式系統的服務。Globus Toolkit (GT)，是 Globus 所提出的工具，大量的整合 Web Service 的機制，利用其相關的協定來定義其元件的架構，並整合其模組的功能，持續的由社群來更新與修定其內容。

Globus Toolkit，提供了一套建構 Grid 平台的解決方案，目前已經到第四版，Globus Toolkit version 4 (GT4)的元件，以功能分類可分成 5 種，分別為：

(1) 安全管理(Security)：安全管理工具可證實用戶的身分或服務的建立，保護用戶的通信，以及確定用戶的授權，並且提供用戶證書管理和維護成員的信息等有關的功能。
(2) 資料管理(Data Management)：資料管理工具，是針對分散式資料的位置與傳輸等相關的管理。
(3) 執行管理(Execution Management)：GRAM 是 Globus 的專案，可提供用戶在以網格為基礎的計算資源上，達到定位(Locate)，提交(Submit)，監測(Monitor)和取消作業(Remote Jobs)等任務，並且能夠提供個別的協定與不同的 Batch/Cluster 作業排程進行傳遞。GRAM 能夠達到遠程的(Remote)的執行管理，並且提供可靠的運算、狀態監測、與認證管理。
(4) 資訊服務(Information Services)：監測和探索系統(Monitoring and Discovery System，MDS)是 Globus Toolkit 的資訊服務之元件，在網格中可提供與資訊相關的可用資源。
(5) 共同運行環境(Common Runtime)：是 Globus 的基礎建設運行的環境[6]，共同運行元件(Common Runtime Components)在 GT4 中是用在 Web 和 Pre-web 服務中，提供各類函式庫及作業服務。這些服務能成為獨立的平台，建置在各種的不同的階層(Various Abstraction Layers)中。

歐洲粒子物理研究中心(European Organization for Nuclear Research；CERN) [4,5]，是大型粒子物理學實驗室，成立於 1954 年 9 月 29 日，並且成立了網格計算的發展中心，主持歐洲科學網格計劃(EGEE)、大型強子對撞器 LHC（Large Hadron Collider）網格計算計畫和 CERN 網際網路交換點(CIXP)。目前台灣中研院已參與 LHC Computing Grid 計畫，與其它國家的高能物理研究單位相連接，共同分享 「大型強子對撞器網格計算計劃」的實驗所需的資源[7]。LHC，需要非常大的計算能力來處理大量的資料，因此 Data Grid 因而出現。而且 CERN 所主導建置的「全球網格」系統，成為了繼網際網路(World Wide Web, WWW)後，全球規模最大、全時性、最穩定的新一代的電腦基礎架構。「全球網格」每年可處理超過 15PB（1 PB = 一百萬 GB）以上的數據資料，因此未來可能在商業上或是在學術領域等創造出許多應用的機會。

Data Grid 應具備的幾項功能[8]：

(1) 負載的排程和管理：在 Data Grid 的環境下，可能會有許多用戶同時提交需求的情況發生，因此在分派任務時，需要考慮到個別設備的能力及網路的通訊情況。除此之外，系統負載的管理需要具備能夠支援資源的協同分配和預留，並且在資源失效的時候，提供所需的恢復策略等。

(2) 資料管理：在具有大量資料的 Data Grid 環境下，如何管理這些資料，是重要的議題之一。一個好的管理，能為 Data Grid 帶來更有效率的資料使用。因此，應該具備一些能夠支援在廣域網路中，資料的分發儲存機制，並且能夠根據用戶的使用方式來決定資料的分發方式。

(3) 網格監控：Data Grid 需要有一個可以監控整個 Data Grid 的介面或設備，因為有效的網格監控，可以幫助 Data Grid 的排程與資料管理等，更能調整整體 Data Grid 的性能。

(4) 構造層管理：透過網格的構造層，可以提供後續在網格中建置任何系統時其安全性與資源分配等之討論。因此，需要中介軟體的基本套件，提供網格靈活及有彈性的管理。目前 Globus 所提供的工具，具有動態配置、自動容錯等能力。但是，隨著網格的技術愈來愈進步，這些能力就愈來愈顯不足，因此需要有創新的方法，可以實現自動化的錯誤偵測和隔離、構造層重組、重新進行任務、與系統新增等等功能。

(5) 大量儲存管理：在 Data Grid 的環境下，有許多來自不同地方的設備提供可儲存的空間，以進行資料的儲存與提取。因此，儲存的管理是一項重要的功能，藉此讓資料能更有效率的進行儲存與利用，以提高資料的使用率及存取率。

由 Data Grid 應具備的功能可知，一個可提供檔案有效儲存的架構，讓使用者在進行檔案儲存時，能夠有效的達到讓使用者滿意的儲存結果，是必要的。因此，在本研究中將提出一個以服務為導向的 SOS(Service Oriented Storage)架構，除了可以提供有效的檔案儲存外，並且以最有效率的方式找出合適的儲存節點(Storage Node)，讓使用者在進行檔案儲存時，獲得較佳的儲存效率。

3. SOS (Service Oriented Storage) 架構

為了在 Data Grid 的環境下，達到資料儲存最佳化，因此本研究提出以服務為導向的 SOS(Service Oriented Storage)架構，藉由這個架構能夠快速並有效的選擇一個最佳的 Storage Node，提高在 Data Grid 中儲存資料的整體效能，包括資料儲存的速度及有效的控管儲存資料等。本研究所提出的 SOS 架構，如圖 2 所示。

SOS 架構中包括三大主要的元件，分別為：Storage Node Decided、Storage Agent、與 Node Clustering。當使用者提出儲存需求時，在 I/O Request Queue 中，將包括所有資料儲存的需求。使用者的 I/O Request 按照排程的順序依序進入 SOS 系統，接著進行決定 Storage Node 的程序。本研究所提出的 SOS 架構，將依據使用者提出儲存需求，對 Storage Node 進行選擇及分群等行為，以便於在決定儲存節點及管理節點時更有效率。當 I/O Request 進入 SOS 架構後，詳細流程將於本節各子章節中說明。

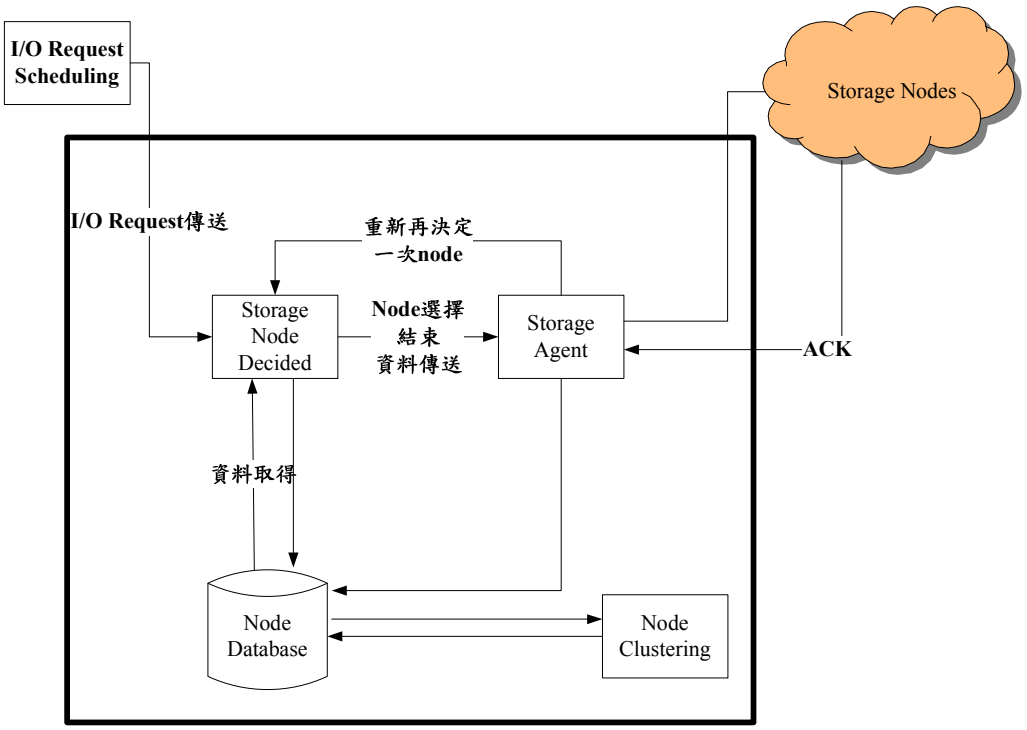

圖 2、SOS 架構

3.1 Storage Node Decided

在 I/O Request 進入 SOS 架構後，第一個會利用到的元件是 Storage Node Decided，本研究將利用 Storage Node Decided 元件找出一個合適的 Storage Node，然後將待儲存的資料傳送到 Storage Agent。

Storage Node Decided 元件細分為 Group Decided 和 Node Decided 兩個子元件，透過這兩個子元件的相輔相成，可以決定適當的 Storage Node，Storage Node Decided 元件工作流程圖如圖 3 所示。3.1.1 及 3.1.2 兩小節中，將分別說明 Group Decided 和 Node Decided 兩個子元件的工作步驟。

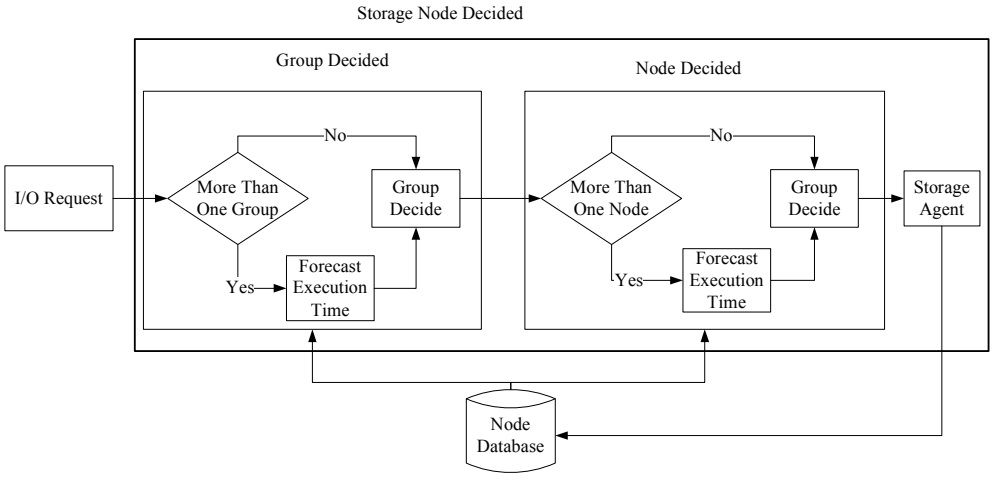

圖 3. Storage Node Decided

3.1.1 Group Decided

當 Group Decided 子元件收到 I/O Request 時，會先取得 I/O Request 的資訊，以便進行 Storage Node Group 的選擇。Group Decided 子元件會從 Node Database 中取得 Storage Node Group 的所有資料，判斷 Data 要儲存到哪個 Storage Node Group 中，如果判斷結果顯示 Data 能夠儲存在一組以上的 Storage Node Group 中，則依照以下步驟決定儲存的 Storage Node Group：

(步驟1) 選擇資料大小與 Storage Node Group 內 Storage Node 大小最接近的群組(Group)，此時有可能會找出超出一組以上群組的。

(步驟2) 估算此資料儲存到每個 Storage Node Group 所需的執行時間。估算評估的因子包括：

- 網路傳輸速度。
- 網路頻寬。
- Data Size。
- Storage Node Group 中，Storage Node 的平均處理時間。

依照以上的步驟流程，可以選擇出一個合適的 Storage Node Group。Group Decided 階段結束後，Storage Node Decided 將進入 Node Decided 階段。

3.1.2 Node Decided

當 Group Decided 決定出一個 Storage Node Group 後，會將訊息傳送到 Node Decided 子元件中。接著，Node Decided 子元件會從 Node Database 中取得由 Group Decided 子元件所選定的 Storage Node Group 中所有目前可用的 Storage Nodes 的資訊，以便決定資料將儲存在那個 Storage Node 中。

在 Group Decided 的階段將避免選擇到無法提供可用 Storage Node 的 Storage Node Group，但是如果發現此 Storage Node Group 的 Storage Node 無法提供儲存服務時時，便會將 I/O Request 回傳到 Group Decided 重新決定 Storage Node Group，同時在 Node Database 將“選擇失敗”訊息加註在其中。

當確定由 Group Decided 子元件所選定的 Storage Node Group 中有 Storage Node 可提供儲存時，將執行下列步驟：

(步驟1) 估算儲存到 Storage Node 可能花費的時間。估算評估的因子包括：

- 每個 Storage Node 的處理速度。
- 網路傳輸速度。
- Data Size。

(步驟2) 決定 Storage Node。

(步驟3) 在選定 Storage Node 後，將在 Node Database 中註記目前此 Storage Node 已經有多少容量被預計用來儲存資料。

(步驟4) 將 I/O Request 的訊息送至 Storage Agent 訊息內容包括：

- I/O Request 內含的資訊。
- Storage Node Group 及 Storage Node 的資訊，例如：Storage Node 所在的位置等訊息。

當完成上述 Group Decided 與 Node Decided 兩個子元件流程後，將進入下一個流程進行資料儲存。

3.2 Storage Agent

本研究將 Storage Agent 元件分為 Data Storage、Node Information Obtain、與 Node Resource Obtain 等三個子元件，Storage Agent 元件如圖 4 所示。3.2.1、3.2.2 及 3.2.3 三小節中，將分別說明 Data Storage、Node Information Obtain、與 Node Resource Obtain 等三個子元件的工作內容。

3.2.1 Data Storage

Data Storage 子元件的功能主要是送出可儲存的訊息給系統選出的 Storage Node。當 Storage Agent 收到 I/O Request 相關資料後，Data Storage 就會送出“檔案可儲存”的訊息通知 Storage Node。然後在 Node Database 中記錄一個“Check”，作為儲存資料的註記。

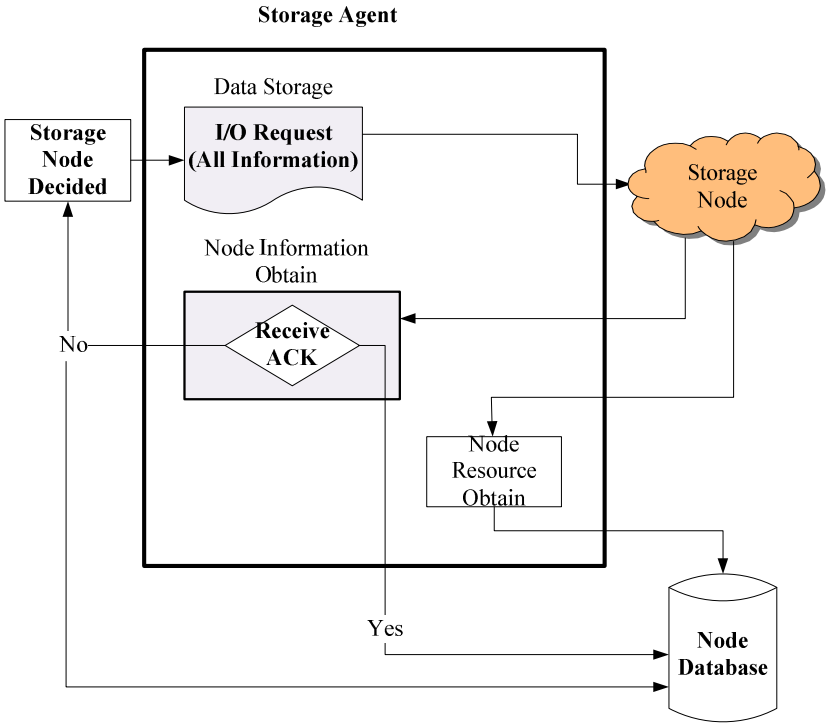

圖 4. Storage Agent

3.2.2 Node Information Obtain

Node Information Obtain 子元件的功能主要是接收 Storage Node 的 ACK，系統藉以判斷儲存結果是否完成。Node Information Obtain 子元件將在特定的時間內收到由 Storage Node 回傳的 ACK。等待回傳 ACK 時可能會發生下列兩種情況：

(1) 收到回傳的 ACK

當 Node Information Obtain 子元件收到由 Storage Node 回傳的 ACK 時，會更新 Node Database 的資訊，其內容包括目前 Node 的可用容量以及 ACK 資料。

(2) 未收到回傳的 ACK

當 Node Information Obtain 子元件在時間內未收到 Storage Node 回傳的 ACK，則會判定儲存失敗。I/O Request 將回傳到 Storage Node Decided，並重新選擇 Storage Node Group 和 Storage Node，並將未收到 ACK 的資訊記錄在 Node Database 中。因為此 Storage Node 可能已經失效了，所以未來在進行 Node Decided 時，此 Storage Node 將不再納入考慮考慮。

3.2.3 Node Resource Obtain

Node Resource Obtain 子元件的功能主要是取得 Storage Node 的資訊。所獲得的資訊將提供 Node Clustering 元件，以 Storage Node 的特性進行分群。因此，當 Node Resource Obtain 子元件收到 Storage Node 的資料時，會將此 Storage Node 的相關資訊儲存在 Node Database 中。

3.3 Node Clustering

本研究將 Node Clustering 元件分成兩種機制，分別為 Node Initial Clustering 和 Node Re-clustering 機制，Node Clustering 元件如圖 5 所示。在 3.3.1 及 3.3.2 兩小節中，將分別說明 Node Initial Clustering 和 Node Re-clustering 的運作機制及方法。

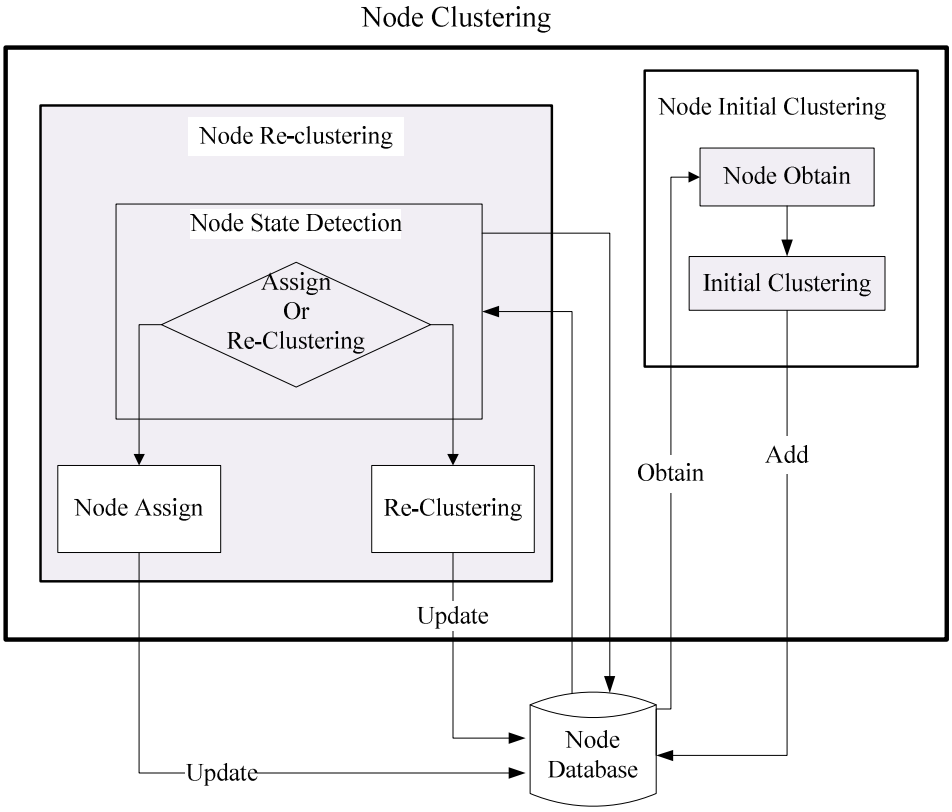

圖 5、Node Clustering

3.3.1 Node Initial Clustering

Node Initial Clustering 機制主要是依 Node Resource Obtain 子元件所取得的 Storage Node 資訊，依 Storage Node 的特性進行分群。透過分群，可以將特性類似或相同的 Storage Node 歸於同一 Storage Node Group。因此，當使用者提出儲存的 I/O Request 時，Group Decided 子元件可以依使用者之 I/O Request 的需求內容，迅速的找到最適當的 Storage Node Group，進而由 Node Decided 子元件決定儲存的 Storage Node。Node Initial Clustering 機制的運作機制如下步驟：

(步驟1) Node Obtain：由 Node Database 中取得所有 Storage Node 的資料。

(步驟2) Initial Clustering：將取得的 Storage Node 資料，依據評估因子進行 Storage Node 的初始分群，評估因子包括：

- Node 的能力，如處理器的速度。
- 檔案傳輸率，如網路頻寬。
- Node 的儲存空間。
- 長駐型 Node 或非長駐型 Node。

當 Node Initial Clustering 機制依 Storage Node 的特性進行分群後，所形成的 Storage Node Group 將儲存在 Node Database 中。

2.3.2 Node Re-clustering

當已完成分群的 Storage Node Group 中的 Storage Node 因為某些因素，造成可用 Storage Node 的能力(包括處理器的速度、檔案傳輸率、儲存空間等)小於 Storage Node Group 所允許的最低限度，在這種情況下就必需考慮要進行重新分群。因此，Node Re-clustering 機制，是繼 Node Initial Clustering 階段後，提供重新分群的功能。重新分群有兩種可能的情形：

(1) **Storage Node 直接分派**：當 Storage Node 可以被其它 Storage Node Group 合併時，則不考慮成本花費較高的重新分群，而將 Storage Node 直接分派給該 Storage Node Group。在這種分派狀況下，I/O Request 的服務就不需停止，仍然可以繼續 Storage Node Decided 及 Storage Agent 的流程。

(2) **Storage Node 重新分群**：Storage Node Groups 中的 Storage Node 已經不敷使用，或是 Storage Node 移出率太高，造成許多 Storage Node Group 中的 Storage Node 無法提供可用的儲存容量時，Storage Node，就必需先暫停 I/O Request 服務的流程，進行 Storage Node 的重新分群。

Node Re-clustering 機制的運作如下：

(步驟1) Node State Detection(節點狀態偵測子元件)：在節點狀態偵測子元件中，會先記錄目前已產生異狀的 Storage Node，並隨時監控這些異狀 Storage Node，當它回復正常時，會將此記錄刪除；如果一直保持異狀的狀態，則仍然保持記錄。節點狀態偵測子元件在記錄 Storage Node 的相關資訊時，會依照 Storage Node 的 Storage Node Group 進行分類記錄，並暫存在 Node Database 中。當節點狀態偵測子元件的判斷機制決定必需要進行 Storage Node 的分派或是重新分群時，就會分別選擇啟動 Node Assign 或是 Re-Clustering。

(步驟2) Node Assign(節點指派)：假設每個 Storage Node Group 的 Storage Node 數，最少需 6 個，最多不可超過 12 個。Node Assign 子元件將找出最接近的 Storage Node Group 條件，將這些 Storage Node 加入那些 Storage Node Group 中。節點分派可分成兩種情況，以舉例來說明。

例 1： 假設 A Group 中目前可用的 Storage Node 數剩餘 10 個，B Group 中目前可用的 Storage Node 數剩餘 5 個，C Group 中目前可用的 Storage Node 數剩餘 6 個，經由 Node Assign 判斷後，若 B Group 中的 Storage Node 與 A Group 的 Storage Node 的特性最接近，而目前 A Group 中的 Storage Node 數並未達到上限時，Node Assign 就會將 B Group 中的 Storage Node 合併到 A Group 中，組合成新的 Storage Node Group。但因合併後其 Storage Node 的個數超過了所制定的 Storage Node Group 之上限數量，因此將合併後的 Storage Node Group 進行分裂。此時，合併後的 Storage Node 將被平均分配到兩個 Storage Node Group 中儲存，並更新 Node Database。

例 2： 如果 B Group 中的 Storage Node 與 C Group 的 Storage Node 的特性最接近，而 B Group 與 C Group 中的 Storage Node 數少於 6 個，也就是未達最低限制的 Storage Node 數時，即合併 B Group 和 C Group，組合成新的 Storage Node Group，並更新 Node Database。

例 3： 然而，若 A、B、C 三個 Group 的 Storage Node 都未達上限，則進行重新分群並重新制定其 Storage Node Group 的範圍。重新分群將在步驟 3 詳細介紹。

(步驟3) Re-Clustering(重新分群)：由於在進行重新分群時，必需停止 I/O Request 的服務流程，等待分群完成後，才能繼續進行 I/O Request 的服務。然而，因為重新分群必須花費非常高的成本和時間，並且會造成 I/O Request 服務運作流程的停擺，因此除非有必要，否則宜盡量避免重新分群的機制啟動。由於，本研究假設 Storage Node 具有高可信度與高穩定等特性，因此 Re-Clustering 機制將不會輕易執行。Re-Clustering 機制包括兩個子步驟：

(步驟3.1) Node Obtain；由 Node Database 中取得所有

Storage Node 的相關資料。

(步驟3.2) Re-Clustering：在取得所有 Storage Node 的相關資料後，依據評估因子進行 Storage Node 的分群，評估因子與 Node Initial Clustering 機制所採用的一致。待分群結束後，將 Storage Node 分群的結果儲存在 Node Database 中。

3.4 Node Database

Node Database 中記錄所有 SOS 架構在運行時所需的資料，包括 Storage Node 的相關資訊，以及一些元件所需的暫存資料。

4. 結論

在本研究中，我們提出了以服務為導向的 SOS(Service Oriented Storage)架構，此架構是由三個不同的元件組成，包括：Storage Node Decided、Storage Agent、與 Node Clustering，除能有效的管理 Storage Node 的使用外，並能掌握目前可用的 Storage Node 的相關資訊。除此之外，SOS 架構更可依 Data Grid 的實際環境，判斷 Storage Node Group 是否需要進行重新分群，並能在 Data Grid 環境下有效的找到適當的 Storage Node，以進行資料的儲存，讓使用者能夠得到滿意的儲存結果。

在未來研究，為使 Storage Node Group 中的 Storage Node 能提供更有效率的服務，我們將進行 Storage Node 動態分群機制的研究，並針對不同的分群機制進行比較，以找出最佳的分群分法。

誌謝

這篇論文是國科會計畫(NSC95-2221-E-324-023)研究成果的一部份。我們在此感謝國科會經費支持這個計畫的研究。

參考文獻

[1] 陳麗娟、肖攸安，“網格計算技術及其應用，”***武漢理工大學學報信息與管理工程版***，27 卷 5 期，2005。

[2] “世界最大規模「網格計算」網路正式啟動，”***北京新浪網(2008)***，Retrieved 2008.12.12，from http://news.sina.com.tw/article/20081004/917189.html。

[3] “About the Globus Toolkit，”***Globus***，Retrieved 2008.12.12，from http://www.globus.org/toolkit/about.html。

[4] “歐洲核子研究組織，”***維基百科***，Retrieved 2008.12.12，from http://zh.wikipedia.org/ wiki/歐洲核子研究組織。

[5] 林誠謙 (2008)，“全球網格正式運轉慶祝典禮(LHC Grid Fest)台灣同步連線共創人類資訊新高峰，”***中央研究院***，Retrieved 2008.12.12， from http://db1n.sinica.edu.tw/textdb/gatenews/showpost.php?rid=1785。

[6] 吳永和、肖君、王雁林，“基於資料網格的教育資源服務系統的構建，”***中國電化教育，華東師範大學網路教育學院及現代遠端教育研究中心***，2005。

[7] 張傑生，“網格運算服務介紹，”***臺灣大學計算機中心***，Retrieved 2008.12.12，from http://grid.ntu.edu.tw/html/intro2.html。

[8] “歐洲資料網格 DataGrid 介紹，”***51CTO.COM***，Retrieved 2008.12.12，from http://tech.51cto.com/html/2006/0322/24178_1.htm。