

## 第四章

### 信任模型設計

在這一章主要是介紹本論文在設計新信賴模型時所使用的公式以及演算法。主要分為兩個部分，第一部分我們將介紹新信賴模型的設計目標；第二部份則進一步地探討新信賴模型所使用的公式與演算法。

#### 4.1. 設計目標

本論文旨在建立一個新的信賴模型，使所有參與其中的電腦節點，能自主且誠實地提供本身的計算資源，進而帶動整體公共生產力向上提升。是以本信賴模型有下列幾項主要的研究目標：

- 1) 建立合理的懲罰制度：本文第二章所提到的兩個信賴模型都認為，當信賴模型中出現詐欺行為後，懲罰值應該“平均分給所有的節點”。但本論文認為合理的懲罰制度應該依照上一章我們所提到的人際關係環境中的「關係序位區」的距離遠近來按照比例給予懲罰值。選定一個節點當作自身節點並要求服務，則懲罰值應該由服務提供節點沿著「信賴路徑」（由推薦的節點連結而成的路徑，例如，若節點 A 為服務要求者，節點 D 為服務提供者，其中節點 C 推薦 D，節點 B 推薦 C，而 B 為 A 的好友，則 A 到 D 之間所連成的路徑稱為信賴路徑），按比例遞減並分配給路徑上的節點。這樣的作法是因為第一序位區所推薦的節點要比第二序位區的節點負起更大的「推薦責任」，也就是因為距離為第一序位區的兩個節點彼此最為熟識，第二序位區的節點則是因為認識朋友才間接知道有被推薦的節點這個個體，所

以責任便會隨著序位區較遠而較小，以此類推。當沿著信賴路徑回到自身節點的好友身上時，責任最小，懲罰也最小。懲罰的比例我們會在稍後的公式與演算法之中做說明。

2) 建立轉手路徑 (Pass-over path)：另一項本論文所提出的新信賴模型與其他信賴模型不同之處，是本文將模型中的每一件計算任務均賦予「成本」的概念。這裡所謂的成本乃狹義地指計算任務「允許多久計算完畢」，也就是計算任務僅僅可以經過多少節點的推薦便需要找到適切的服務提供者，否則將宣告計算任務失敗。當計算任務從要求服務的自身節點出發開始，成本便往上累加；當找到適當的服務提供節點時，成本便會停止累加，但剩餘的成本會記錄在服務提供者身上。當計算任務交付給服務提供者後，表示信賴路徑已經形成，信賴路徑負責找到適當的服務提供者，並將計算任務排入服務提供者的「服務佇列」(Service queue)中，等待被服務。我們定義「適當的服務提供者」乃是狹義的以服務提供者在被詢問的當時，服務提供者具有符合服務要求條件的服務、負載情況在每一個個體自己定義的可服務範圍內(本論文在信賴模型中將賦予每一個節點一個隨機的負載情況，以模擬真實的電腦節點)，且服務佇列尚有空位。由於服務佇列中每一個計算任務所使用的成本均不相同，因此，若依照使用成本的多寡來排列服務佇列中的計算任務的優先順序，就會出現有些計算任務無法在預期的時間內完成，這時候便需要向外尋找轉手的對象，以期其他擁有相同服務且信譽良好的節點可以提供協助。我們定義，「從被賦予計算任務的節點開始，到可提供運算協助且信譽良好的節點為止」，這一段連結起來的路徑，我們稱為轉手路徑。轉手路徑的目的在於減少計算任務失敗的比例以及讓同樣提供優良服務，但卻尚未建立起口碑的節點一樣有機會得到計算任務，而不會長期處於飢餓狀態。形成轉手路徑所需要的明確條件我們會在稍後的公式與演算法之中做說明。

3) 建立同業公會 (Guild) 制度：本文前面所提兩個信賴模型，仍僅採用可分享的運算資源統一向伺服器或是集線器登記註冊的方式來管理所有的運算資源。以 D-S 證據推論信賴模型為例，當一個新的節點要進入分散式分享網路時，便是向鄰近的集線器登記自己可分享的運算資源。這樣的作法可能會

造成當新加入的服務節點  $A$  所屬的集線器出現當機或是網路失效時，節點  $A$  必須重新向其他集線器做登記才有可能在下一次的服務需求之中被賦予計算任務。再者，新節點  $A$  一開始被賦予的信譽值都比較低，除非同一個集線器所管轄的其他服務節點都出現熱點現象，迫使模型將計算任務賦予給新節點  $A$ ，否則  $A$  還是有很高的機會出現飢餓的情況，信譽值也不易提升。

因此本論文提議建立一個「同業公會」制度，所謂的同業公會是指擁有相同服務的電腦節點有一個可以認識彼此的「場所」（我們定義所謂的場所是指每一個節點均具備一個叫做 guild list 的 queue，可以記錄曾經認識的提供相同服務的節點）。以上述新加入的節點  $A$  為例，當  $A$  剛進入小世界環境時， $A$  將會「主動且隨機」地挑選小世界中的一個節點  $B$  作為社交對象，並交換彼此的分享資訊。交換的過程中倘若節點  $B$  恰巧與  $A$  提供相同的服務，則  $A$  可以得到  $B$  所認識的同業公會的節點名單（Guild list），並且以信賴模型所賦予新節點的預設信譽值排入  $B$  的好友名單以及同業公會名單的最末端， $B$  也將成為  $A$  的第一個「好友」；若  $A$  與  $B$  提供不同服務，則  $A$  仍依照信賴模型所賦予新節點的預設信譽值排入  $B$  的好友名單的最末端，且  $B$  也將成為  $A$  的第一個好友。

新加入的節點採取「主動且隨機」的方式認識小世界裡的節點，並藉機建立起同業公會名單，是本論文所提出的新信賴模型與其他信賴模型地另一項不同處。我們認為以行動理論為基礎建立的信賴模型，每一個參與其中的節點均受到「取得計算任務並提高自身節點的信譽值」的欲望所驅使進而跟小世界產生互動，並且藉此認識第一個好友甚至進入好友的同業公會之中，增加自身節點的曝光率。待新加入的節點擁有一份「部份」（每一個節點所擁有的同業公會名單不盡相同，但在信賴路徑或是轉手路徑的建立過程中會慢慢的增加）的同業公會名單後，表示自身節點也在他人的同業公會名單中，如此一來便可以在好友發生轉手路徑時，有機會從好友得到計算任務，進而提高自身節點在小世界中的聲望。同業公會機制可以讓本文所提的新信賴模型較其他模型來得更強壯（Robust），不會因為登記的集線器或是伺服器發生問題使得所屬的分享運算資源的節點必須再向其他集線器或節點登記才不至於無法接受計算任務；更可以因為符合人際環境關係，而使得懷

有惡意的節點無法針對某個登記的集線器實行詐欺手段取得計算任務，善意且優質的節點也不用單方面的等待計算任務的賦予，可以主動的提升自身節點在小世界中的聲望。

我們的設計目標在於更明確的將社會方法論/行動理論以及人際環境關係理論落實在本論文所提出的新信賴模型之中，並藉以改進我們認為其他信賴模型不足之處。我們將在下一節詳細探討我們所應用在模型中的公式以及演算法。

## 4.2. 本論文所提出之新信賴模型所定義的公式與引用的演算法

我們假設一條簡單且完整的信賴路徑，至少包含在同一時間各自扮演不同身分的三個節點：服務提供者、為服務要求者介紹優良服務的掮客節點（Broker node）、以及服務要求者。我們假設服務要求者  $A$  被賦予一個新的計算任務  $X$  並產生服務需求，則節點  $A$  透過可信賴節點  $B$ （即節點  $B$  為節點  $A$  的好友之一），詢問可提供公用計算服務  $X$  的電腦節點。我們假定信賴度為一個小於 1 大於 0 的非負浮點數  $c$ ：  $0.00 \leq c \leq 1.00$ ，且在詢問時間  $t$  內（我們視為瞬間），信賴度並不會改變。假設此時恰有一節點  $C$  可提供計算服務  $X$ ，而  $B$  對其信賴度為  $c_{bc}(t)=0.85$ ， $A$  對  $B$  的信賴度為  $c_{ab}(t)=0.76$ ，則我們可以知道  $A$  對  $C$  的信賴度（如圖 4.1） $c_{ac}(t)$  可以寫成下列數學式：

$$c_{ac}(t) = c_{ab}(t) \times c_{bc}(t) = 0.76 \times 0.85 = 0.65 \text{ (四捨五入)}$$

所以我們可以得到任一節點  $i$  對節點  $j$  的公式：

$$c_{ij}(t) = c_{i1}(t) \times c_{12}(t) \times c_{23}(t) \times \dots \times c_{(n-1)n}(t) \times c_{nj}(t) \quad \dots (1)$$

其中，節點  $1 \dots n$  為節點  $i$  到  $j$  的信賴路徑上所經過的節點。

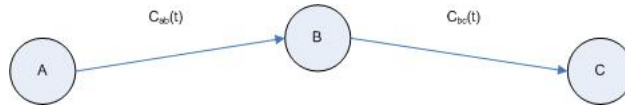


圖 4.1，簡單且完整的信賴路徑。

我們考慮複數信賴路徑存在的可能性，也就是節點  $i$  對節點  $j$  間信賴路徑的個數為大於或等於 1 的情況（也就是信賴路徑至少會有一條）。我們定義  $A$  綜合評量（平均） $C$  的信賴度為  $c_{average}(t)$ ：

$$c_{average}(t) = 1/n [ c_{ij1}(t) + c_{ij2}(t) + \dots + c_{ijn}(t) ] = 1/n \sum_{k=1 \dots n} c_{ijk}(t) \quad \dots (2)$$

其中， $c_{ijk}(t)$  表示  $n$  條信賴路徑中第  $k$  條信賴路徑上節點  $i$  對節點  $j$  的信賴度。

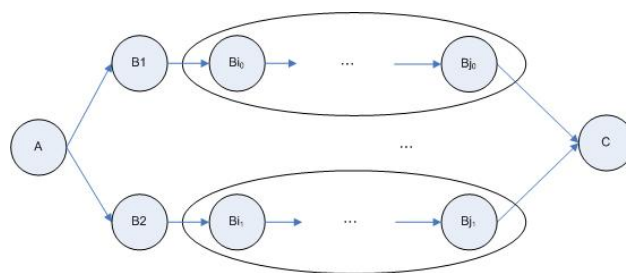


圖 4.2，複數信賴路徑。圖中兩段用橢圓形圈起來的路徑區段可以完全不重複、部分或全部重複。

上述兩個基本公式即為本信賴模型的信賴關係建立的基礎。我們可以看出信賴路徑的建立其實是一種競爭關係，好友名單中的每一個節點以信譽值排行做為競爭，希望可以提供服務提昇聲望。我們假設在競爭正常發生的情況下，信譽值會隨著競爭後的好壞而有口碑的增減；於是好將會更好，而服務品質不佳或是惡意破壞整體小世界架構的節點，也都會很快的被淘汰。

本論文提出的新信賴模型除了競爭關係外，另一個重要的互動關係便是合作。與競爭關係不同，合作關係區分為兩個部份：同業公會的形成以及建立轉手路徑。我們假設新的計算節點在加入時，將主動並隨機挑選小世界中的任一節點作為社交對象，加入好友名單之內，並且交換分享資訊。倘若此一隨機挑選的節點正好提供相同的服務，則新加入的節點將會得到一份一模一樣的同業公會名單，而本身也將依照預設的信譽值成為被訪問節點同業公會名單中的一員，這就是所謂的「同業公會機制」。同業公會的機制最主要的作用在於讓新加入的節點有機會很快的融入小世界的運作之中，並且在友好節點必須將計算任務分享時，有機會得到計算任務，讓整個信賴模型飢餓的機率大幅降低。

由於合作關係的失敗與否將關係到一個以上節點的信譽值，所以在我們定義轉手路徑之前，我們將先定義合作關係中信譽值的增減。

我們認為合作關係的成功會讓服務提供者在服務者要求心中的信譽值上升；反之若是計算任務失敗則信譽值將會降低。我們假設信譽值的增加為前一個瞬間的信賴值加上此次增加的信賴量。假設節點  $j$  在時間  $t_0$  時對  $i$  而言信賴度為  $c_{ij}(t_0)$ ，信賴度變化量為  $\delta$ ，則我們可以定義合作成功與失敗所造成的信賴量變化為  $\pm\delta$ ， $0.00 \leq c_{ij}(t_0) \leq 1.00$ ，此時當合作的計算任務成功與失敗時的信賴度  $c_{ij}$  可以用以下的數學式計算得知：

$$c_{ij}(t_1) = c_{ij}(t_0) \pm \delta \quad \dots (3)$$

我們假設，當某一節點  $i$  擁有計算任務  $X$  並尋求服務提供者，在詢問過好友名單中所有信譽值在某信心強度  $CI$  ( Confidence Intensity ) ( 此為一可設定參數 ) 之上的信賴節點後決定採用信賴評價最高且提供所需服務的節點  $j$  的服務。此時，節點  $i$  便會將排入  $j$  的工作佇列最後，並請求  $j$  的支援。我們定義計算任務  $X$  將在節點  $j$  的工作佇列中被排序，倘若排序後計算任務  $X$  的剩餘工作成本仍小於排在  $X$  前面的工作數量，表示節點  $j$  無法在規定時間之內完成計算任務  $X$ 。這時候  $j$  為了保有自身節點的聲望不致下降，就必須向外請求同業公會中的好友支援，於轉手行為開始發生，轉手路徑開始形成。

計算任務的轉手，通常意味著額外工作成本的增加。我們假設以節點  $P_1$  為起點的轉手路徑，當  $P_1$  轉手給  $P_2$ ， $P_1$  便會受到一次信賴增量打折率的效果；當  $P_2$  轉手給  $P_3$ ， $P_2$  就會受到一次信賴增量打折率的效果，而  $P_1$  則會再受一次信賴增量打折率的效果，以此類推。信賴增量打折率的累乘表示即使此計算任務最終在轉手路徑上的第  $n$  個節點完成，此時轉手路徑上第  $m$  個節點  $P_m$  也只能得到  $\delta \times (\text{信賴增量打折率})^{(n-m)}$  的信賴增量，而非完整的  $\delta$ ，只有完成任務的第  $n$  個節點有資格得到  $\delta$  的信賴增量。我們定義信賴增量打折率為  $\Delta$ ，則我們可以得到公式 (4)：

$$\delta_{P_m} = \delta \times \Delta^{(n-m)} \quad \dots (4)$$

其中， $n$  表示轉手路徑上成功完成計算任務的第  $n$  個節點。 $1 \leq m \leq n$ 。

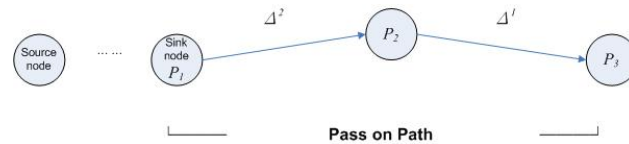


圖 4.3，轉手路徑。

同樣的，若計算任務在第  $n$  個節點宣告失敗，第  $n$  個節點亦應負起計算失敗的全責，如同沒有發生轉手現象而使計算任務失敗的節點一樣，失去一個完整的信賴增量，而轉手路徑上的任一節點  $m$  亦因任務的失敗，受到失去  $\delta \times (\Delta)^{(n-m)}$  的信賴量懲罰。是以要求服務節點  $i$  對任一轉手路徑上的節點  $j$  的信賴關係可以寫成下面的公式：

$$c_{ij}(t_1) = c_{ij}(t_0) \pm \delta_{Pm} \quad \dots (5)$$

透過上述的設計目標，我們預期整個新信賴模型能讓服務要求者的計算任務可以取得服務品質良好的計算服務，也可以降低被惡意詐欺或破壞的機率。提供服務者也不會因為其中幾個少數較熱門的節點形成為熱點，而出現單點崩潰的疑慮，也不會使新加入或仍未被其他同一網路群組的電腦所熟悉的計算節點，長時間處於服務飢餓（Starvation）的狀態，永遠或是鮮少有計算任務登門，浪費公用計算資源。在下一章，我們將設計一個簡單的模擬器，來驗證本文的論點。