無線分散式計算網路之設計與分析

(Design and Analysis of Wireless Distributed Computing Networks)

摘要

在這個資訊化的時代下，由於許多應用程式都需要巨大的計算能力才能夠完成，分散式計算網路 (distributed computing networks) 進而受到注目，它研究如何把一個需要巨大計算能力才能解決的問題分成許多小的部分，並分配給許多電腦進行處理，且整合得到最終的結果。然而，隨著科技的迅速發展以及通訊科技趨於無線化，促成了電腦網路一項重要領域 "**無線分散式計算網路** (wireless distributed computing networks)" 的崛起，此領域為本計畫研究的重點亦為未來的發展趨勢，我們深入研究無線分散式計算網路，設計動態封包排程 (dynamic packet scheduling) 有效地分配網路資源，並且同時達到"任務完成時間 (task completion time)"的最小化，我們對此方面產生興趣，並且深入探討其中的方法，期待在本計畫能夠做出一個令人稱奇的傑作。

(一)研究動機

近年來，越來越多的科技都依賴著龐大的運算能力才足以完成，如深度學習 (deep learning)、服務導向架構 (service-oriented architecture) 等等。為了達到這些技術所需的運算量，分散式計算網路因而受到重視。其概念在於將所需的運算拆解成許多的小量運算，並分配給多部伺服器 (server) 同步進行處理，以降低總共所需的運算時間。此外，隨著通訊科技趨於無線化，**無線分散式計算網路** (wireless distributed computing networks) [1,2] 的開發逐漸受到重視，亦是未來分散式計算的趨勢 [1]。因而，本提案將針對無線分散式計算網路，探討其所帶來的挑戰，並提出未來的研究方向 。

在過去傳統網路的設計中，設計者多把焦點著重於"網路吞吐量"的最大化、亦或"封包延遲"的最小化。然而，由於分散式計算網路的目的在於降低運算的時間，因此，如何設計一個網路以達到"任務完成時間" [1] 的最小化，將是分散式網路設計的重點。此外，在無線網路中，無線伺服器彼此間的訊號容易相互干擾，因而帶給無線分散式計算網路設計者許多新的挑戰 [1]。因此，在本計畫中，我們將研究如何在無線分散式計算網路中設計動態的封包排程以避免干擾，並同時達到任務完成時間的最小化。

(二)研究問題描述

為了深入研究無線分散式計算網路的動態封包排程，我們將考慮一個相關文獻 [3] 所提出的簡化系統以及“coflow”的概念。我們以圖1為例，說明coflow的定義以及未來將探討的問題：

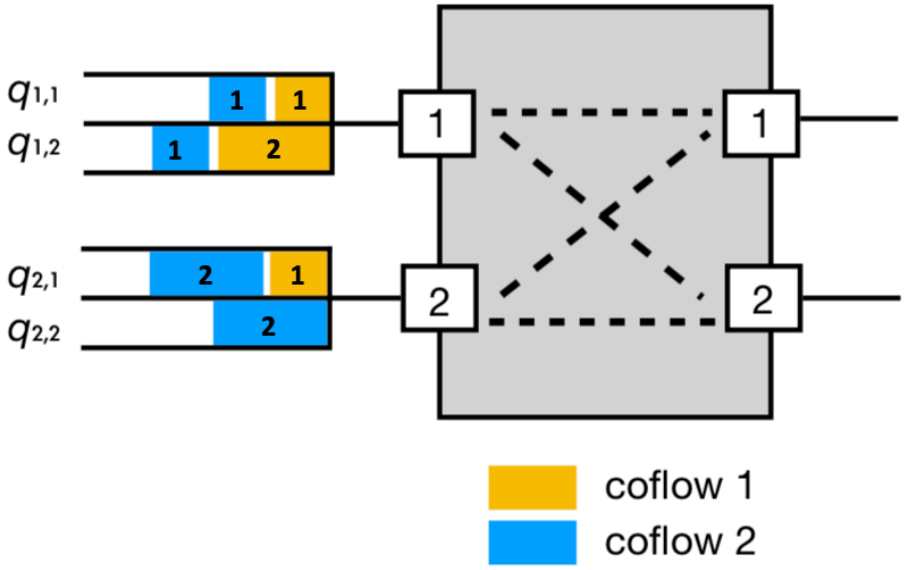

Figure 1： 2×2 輸入排隊交換機與coflow

- 圖1為一個由兩個輸入端口與兩個輸出端口所組成的輸入排隊交換機 (input-queued switch)。左邊的每個輸入端可連結到右邊所有的的輸出端。此外，針對每個輸入－輸出的組合，輸入端設置一個佇列 (queue) 來存放即將送往相對應輸出端的封包，即圖中的$q_{1,1}$，$q_{1,2}$，$q_{2,1}$，$q_{2,2}$。事實上，這樣的輸入排隊交換機已函蓋了許多分散式計算網路，以MapReduce為例，如圖2，我們把每個端口視為一個無線伺服器，其中左邊的輸入端口為mappers，而右邊的輸出端口則是reducers。此外，這樣的輸入排隊交換機亦包含了許多無線通訊網路系統，如下面表格所示：

通訊系統	輸入排隊交換機
點對點通訊	1 輸入端口x 1 輸出端口
廣播通訊	1 輸入端口x M 輸出端口
聚合通訊	N 輸入端口x 1 輸出端口
多個使用者共享	N 輸入端口x M 輸出端口

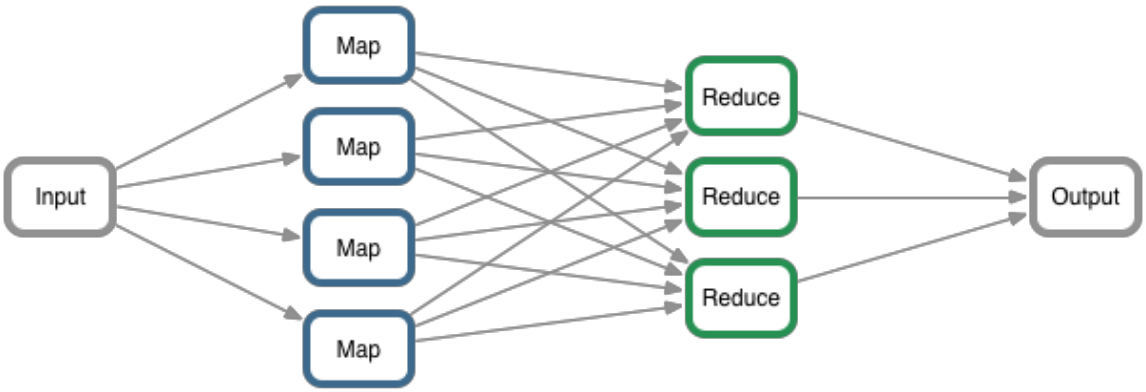

Figure 2 : MapReduce 系統

- 接下來，我們定義coflow。一個coflow指的是一組屬於同一個任務的工作 (task) 封包。為了方便解釋，我們假設在圖1的輸入排隊交換機操作於離散時槽系統 (discrete slot system)。在第一個時槽，有一個coflow（以橘色表示）到來，其中該coflow的任務包含了三個工作封包分別放置於$q_{1,1}$，$q_{1,2}$，$q_{2,1}$，其工作封包大小分別為1個單位、2個單位、1個單位。換句話說，在每一時槽中，可能有一coflow到來，其中coflow所屬任務裡的每個工作封包大小可以是隨機的。

- 我們假設圖1裡的輸入排隊交換機裡，每個輸入端口在每個時槽中只能傳輸出至一個輸出端口，且每個輸出端口在每個時槽中也只能接受一個輸入端，這通常稱之為“交叉開關約束 (crossbar constraints)”。這交叉開關約束是用來描述無線網路裡的干擾情形。根據交叉開關約束的限制，在一個時槽裡有兩種可能的傳輸組合 (而不會造成干擾): $L_1 = \{$輸入端1號 → 輸出端1號, 輸入端2號 → 輸出端2號$\}$, 以及 $L_2 = \{$輸入端1號 → 輸出端2號, 輸入端2號 → 輸出端1號$\}$。換句話說，在每一個時槽裡，只有 L_1或L_2其中之一種組合可被啟動用來傳輸工作封包而不造成干擾。因此，在每一個時槽裡，我們必須決定在該時槽下要啟動L_1亦或L_2。這樣的決策稱之為動態封包排程設計。

- 接下來，我們定義coflow任務完成時間用以當作排程設計的基準。一個coflow的任務完成時間為：該coflow進入輸入排隊交換機與該coflow所屬的工作封包完全送達至輸出端的時間差。亦即該coflow在輸入端佇列裡所居留的所有時間。對應於於過去對於單一封包延遲的定義，由於一個coflow包含多個工作封包，因此其coflow任務完成時間則為其中最長的封包延遲時間。在本計畫裡，我們將設計動態封包排程設計以達到最小的平均任務完成時間，進而達到最高的分散式計算效率。

(三)文獻回顧與探討

在文獻[1]中，作者擴展傳統的分散式計算方法，以允許在動態無線電環境中運行，並在幫助下滿足無線分散式計算網路獨有的設計和實施挑戰可用的支持技術，讓我們了解無線分散式計算網路為未來分散式計算的趨勢並帶給無線分散式計算網路設計者許多新的挑戰。

此外，在文獻[2]中，作者研究了一種無線分佈式計算框架並遵循MapReduce的結構，在一次性線性方案的假設下，描述了計算負載和通信負載之間的基本權衡，讓我們對MapReduce系統內部構甚至是Hadoop分散式檔案系統 (HDFS) 有了足夠的概念，並給了我們對無線分散式計算網路設計的啟發。

最後，在文獻[3]中，作者提出的通訊網路的簡化系統以及“coflow”的概念，考慮N×N輸入排隊交換機，分析了一些現有與coflow無關的調度策略，且表明它們都沒有在“任務完成時間”方面達到可證明的最佳性能，並提出了Coflow-Aware Batching（CAB）策略，該策略在一些溫和的假設下實現了任務最佳完成時間。這篇文獻引導我們許多方向，並給了我們輸入排隊交換機與coflow的概念。

在本計畫中，我們將結合三篇文獻所提出概念，來設計與分析無線分散式計算網路。

(四)研究方法及步驟

在這個章節裡，首先於第4.1章節，我們提出一個初步的模擬結果，其顯示出這個計畫未來的可能性。接著於第4.2章節，我們將提出未來的研究方向以以可行的方法。

4.1 初步研究結果

根據第3章節的問題描述，我們使用電腦程式來模擬兩種不同的動態封包排程演算法，並進行比較。第一種演算法為大家所熟知的最大權重排程演算法(maximum weight scheduling) [4]，此排程演算法已被證實可使得第3章節裡的輸入排隊交換機達到最大的吞吐量。接下來，在本提案中，我們提出一個簡單的的排程演算法當作第二種演算法。經由模擬發現，對比於最大權重排程演算法，我們所提出的簡單想法即可對於平均任務完成時間有所改善。

首先，我們簡述最大權重排程演算法如下。我們定義$Q_{i,j}(t)$為時槽t時佇列$q_{i,j}$ 裡的工作封包數。在時槽t時，若$Q_{1,1}(t)+Q_{2,2}(t)\geq Q_{1,2}(t)+Q_{2,1}(t)$，則最大權重排程演算法在該時槽裡選擇啟動$L_1$。反之，則啟動$L_2$。其概念在於選擇當下最壅塞的佇列。

接著，我們簡述所提出來的想法，以探索改善coflow任務完成時間的可行性。在時槽t時，我們定義$T_{i,j}(t)$為佇列$q_{i,j}$最前頭封包進入該輸入排隊交換機的時槽。那麼，在時槽t時，若$\min(T_{1,1},\ T_{2,2})\leq \min(T_{1,2},\ T_{2,1})$，則在該時槽裡選擇啟動$L_1$。反之，則啟動$L_2$。其概念在於選擇當下等待最久的佇列。

圖3為電腦模擬的結果。在該模擬中，我們假設每個時槽有一個coflow進入輸入排隊交換機的機率固定為0.4。此外，若是該時槽有一coflow進來，該coflow將包含四個工作封包，分別放置於$q_{1,1}$，$q_{1,2}$，$q_{2,1}$，$q_{2,2}$。此外，我們假設每個工作封包所需的傳輸時間為幾何分布，其平均為$1/p$。在圖3中，橫軸為p，縱軸為平均的任務完成時間。由此可看出我們所提出的簡單想法明顯地改善最大權重排程演算法的平均任務完成時間，所以我們認為該研究有極大的可行性。在接下來的章節，我們將提出更多想法，以待接下來完成。

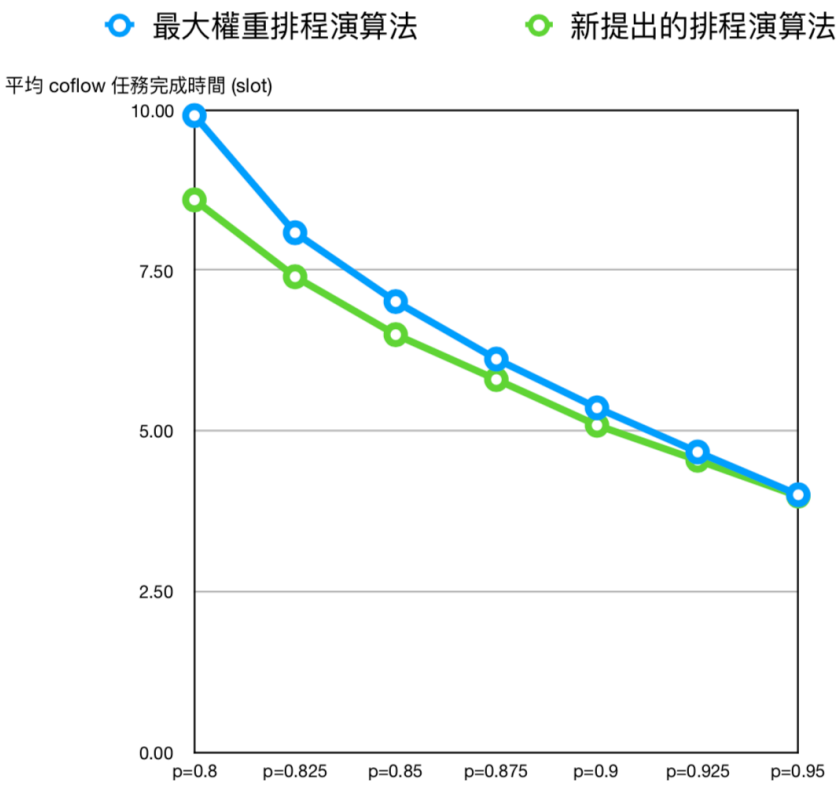

figure 3： 平均任務完成時間比較圖

4.2 未來深入研究方法

在本計畫中，我們將針對輸入排隊交換機，深入研究coflow的動態封包排程的議題，其目的在於達到最小的平均任務完成時間。以下我們提出幾個未來的研究方向：

- 在初步結果中，我們只考慮了$\min(T_{1,1}, T_{2,2})$與$\min(T_{1,2}, T_{2,1})$的比較，亦即只考慮了等待最久的工作封包。或許，我們可以學習最大權重排程演算法的概念，考慮$T_{1,1}+T_{2,2}$以及$T_{1,2}+T_{2,1}$的比較，用以判別啟用哪一組傳輸組合。

- 此外，最大權重排程演算法可藉由Lypaunov theory來證明其吞吐量的最大化，因此，我們將學習其精神來研究所提來的方法在數學上所能保證的效能。

- 除了傳統的通訊傳輸外，我們亦可以考慮加入編碼的概念，以達到更小的任務完成時間。在此，我們簡單的以repetition code當例子，說明其可行性。我們考慮圖的輸入排隊交換機，並假設：

    - 在時槽$t=1$時，有一coflow進入，但只包含一個工作封包在$q_{1,1}$，而其裡佇列

裡並無工任封包，並且之後沒有任何coflow進入；

- 假設，該工作所需傳輸時間（因傳輸通道品質影響）為隨機的，其分布為：需時1, 2, 3個時槽的機率各為1/3。

因此，我們可以計算出該coflow任務完成時間的期望值為： $1 \cdot \frac{1}{3} + 2 \cdot \frac{1}{3} + 3 \cdot \frac{1}{3} = 2$。接下來我們考慮repetition code：將$q_{1,1}$的工作封包複製到$q_{2,2}$。接著，我們可以啟動L_1組合，亦即同時啟用{輸入端1號 → 輸出端1號, 輸入端2號 → 輸出端2號}。因此，$q_{1,1}$及$q_{2,2}$中只要有其中之一個工作封包傳輸完成即完成該coflow的任務。我們令$(\mathrm{X}_1, \mathrm{X}_2)$分別代表$q_{1,1}$及$q_{2,2}$裡的工作封包傳輸時間。我們可分析任務完成時間的機率分布為：

$$\mathrm{P}\{\text{一個時槽}\} = \mathrm{P}\{(\mathrm{X}_1, \mathrm{X}_2) \in (1,1), (1,2), (1,3), (2,1), (3,1)\} = 5/9;$$

$$\mathrm{P}\{\text{兩個時槽}\} = \mathrm{P}\{(\mathrm{X}_1, \mathrm{X}_2) \in (2,2), (2,3), (3,2)\} = 3/9;$$

$$\mathrm{P}\{\text{三個時槽}\} = \mathrm{P}\{(\mathrm{X}_1, \mathrm{X}_2) \in (3,3)\} = 1/9;$$

因此該coflow任務完成時間的期望值為$1 \cdot \frac{5}{9} + 2 \cdot \frac{3}{9} + 3 \cdot \frac{1}{9} = \frac{14}{9} < 2$。根據上述分析可知repetition code可進一步改善任務完成時間。

- 但是，值得我們注意的是，一旦採用了repetition code，則該輸入排隊交換機所需處理的工作封包數量將變多，進而增加了佇列排隊所造成的工作封包延遲。因此，在本計畫中，我們將進一步討論：
  - 使用repetition code對工作封包延遲的影響；
  - 考慮coflow會隨機進入排隊交換機時，該如何妥善使repetition code，問題包括何時要重覆工作封包、要重覆幾個工作封包、要如何放置重覆的工作封包等問題。
  - 除了repetition code之後，我們將考慮其它編碼的可能性，如MDS code等。
- 在本提案中，我們以2×2的輸入排隊交換機為例說明我們提案的可行性，以及未來研究的想法。然而，在接下來的研究中，我們不設限於2×2的輸入輸入排隊交換機，將考慮M×N的輸入排隊交換機，或是更多層（超過2層）的輸入排隊交換機。

(五)預期結果

根據以上的討論，我們期望能在接下來研究可以回答以下重要的問題（但不限制於）：

- 針對輸入排隊交換機，如何開發動態排程演算法，以達到最小的coflow平均任務完成時間；
- 如何加入編碼的概念，進一步改善coflow平均任務完成時間；
- 以電腦模擬研究所提出來的演算法；
- 以Lyapunov theory來研究所提出來的演算法在數學上的效能保證。

(六)參考文獻

[1] Datla, Dinesh, et al. "Wireless distributed computing: a survey of research challenges." IEEE Communications Magazine 50.1 (2012): 144-152.

[2] Li, Fan, Jinyuan Chen, and Zhiying Wang. "Wireless mapreduce distributed computing." 2018 IEEE International Symposium on Information Theory (ISIT). IEEE, 2018.

[3] Liang, Qingkai, and Eytan Modiano. "Coflow scheduling in input-queued switches: Optimal delay scaling and algorithms." IEEE INFOCOM 2017-IEEE Conference on Computer Communications. IEEE, 2017.

[4] Mekkittikul, Adisak, and Nick McKeown. "A practical scheduling algorithm to achieve 100% throughput in input-queued switches." Proceedings. IEEE INFOCOM'98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No. 98. Vol. 2. IEEE, 1998.

(七)需要指導教授指導內容

- 文獻閱讀協助：由於本計畫中諸多參考文獻內容涉及高專業領域之知識，故學生較難獨立理解，需指導教授從旁協助閱讀。
- 數學證明指導：本計畫需要許多相關數學之運算及證明，須由指導教授引導驗證。

- 專業知識辯讀：在解決本計畫問題中，曾有對專業知識的錯誤認知，需由老師在旁引導協助指正。
- 討論延伸問題:由於本計畫中將可能發生預料外的結果，需由老師引導協助解決。