

Multiprocessor Scheduling

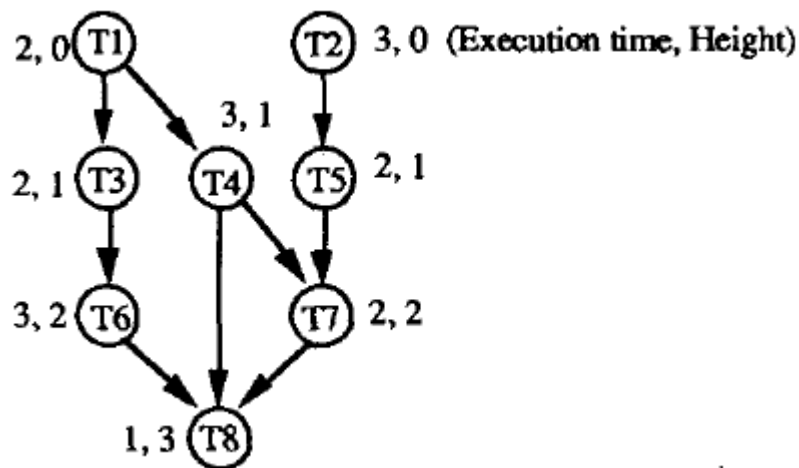
Multiprocessor 的 Scheduling 可依 processor 間的 relationship，進而分為 2 種形式。一種為 Asymmetric Multiprocessing(ASMP)，另一種為 Symmetric Multiprocessing(SMP)。

那簡單來說，這兩者的差異在於前者的 processors 為 Master-Slave 的關係，意指存在一顆主 processor 負責 control 所有 scheduling 跟 I/O processing，而其餘的 processors 聽從他的指揮執行 tasks；而後者的 processors 為自主運行，不存在 Master-Slave 的主從關係。其中，Symmetric Multiprocessing 的 ready queue 又可分為 2 種，一為各個 processor 間的 ready queue 為共享的，另一種為各個 processor 擁有獨立的 ready queue。那 scheduling 又是如何決定的呢？為了瞭解方法，我們找了這篇 paper，”A Genetic Algorithm for Multiprocessor Scheduling-by Edwin S. H. Hou, Member, IEEE, Ninvan Ansari, Member, IEEE, and Hong Ren” 來說明 scheduling 是如何被決定的。

最小化 tasks 在 multiprocessor scheduling 執行的時間是被視為 NP-hard 的問題(difficult to find the answer in polynomial time and cannot guarantee to find the answer in polynomial time)，而這份 paper 中提出一種有效的 **Genetic Algorithm** 來解決 multiprocessor 的 scheduling 問題。

首先其定義 a set of partially ordered 的 tasks 可被表示為

directed acyclic task graph , $TG = (V, E)$ 。其中 $V = \{T_1, T_2, T_3, \dots, T_m\}$, 代表 Vertices , 為 collection of tasks ; $E = \{e_{ij}\}$, 代表 Edges , 其中 e_{ij} 代表 tasks 間的 precedence 關係。上述可以一張圖作為描述:



而上圖中的 height 為該 Task 之 predecessor 中最大的 height+1 , 除非該 Task 沒有 predecessor , 則為 0。直接以數字表達 Tasks 間的順序, 但若在 TG 圖上沒有 Edge 相連的 Task(表 Task 無相依性), 則執行順序可任意安排。

接下來便是Algorithm的部分。Genetic Algorithm可分為以下3個步驟,

- 1 .Initialization – a initial population of the search nodes is randomly generated
- 2 .Evaluation of the fitness function – the fitness value of each node is calculated according to the fitness function(objective function)
- 3 . Genetic operations – new search nodes are generated randomly by examining the fitness value of the search nodes and applying the genetic operators to the search nodes
- 4 . Repeat steps 2 and 3 until the algorithm converges

此Algorithm為一種“世代傳承”的概念, 藉由fitness function, 將好的path迭代至下一個世代且結合不同的path以得出最佳化的結果。

以下分別介紹Genetic Algorithm的3個階段:

- Initialization
 - String Representation : 將Tasks間任意組合成String Space , 並在String Space中挑出legal elements組合成Search Space
 - Initial Population : Search Space
- Fitness Function
 - Used to evaluate the search nodes and also controlling the genetic

operators, which is based on the finishing time of the schedule.

- Genetic Operation

- Crossover : select a crossover site to exchange portions of the strings to get optimize outcome
- Reproduction : string with higher fitness value has higher chance to get into next generation
- Mutation : an escape mechanism for premature convergence

上列步驟可簡化為流程圖:

