

批量作業還是逐個處理

今天，我們的公寓供水出了點問題，暫時無法供水。吃完飯後，看著我和家人留下的一堆碗碟，我想到了幾個問題。

一個是如何讓洗碗機在沒有供水的情況下繼續工作。它可能設計成可以連接到一桶水。此外，水管連接頭應該靈活，以便輕鬆從公共供水切換到私人自製的水桶。

另一個問題是分批還是一個接一個地完成工作。我們可以每頓飯後洗碗，或者等一天或幾天後再洗。這是從我們吃飯和洗碗的角度來看。我們也可以從洗碗機能容納多少碗碟的角度來處理這個問題。

這讓我想到了編程。我們可以分批或一個接一個地執行任務。

分批工作會導致一個明顯的問題：需要更多的資源。因為我們延遲了清洗，所以需要更多的碗碟，並且因為我們積累了數據以推遲處理，所以需要更多的內存空間。

在現實生活中，一次能處理的空間或物品數量是有限的。例如，洗碗機最多可能只能處理二十個碗碟，就像程序在計算機上有內存限制，或者道路有能通過的車輛數量限制一樣。

還有如何分離工作的問題。我們應該一次分離一個項目還是三個項目？

對於碗碟或汽車，將每個物品視為一個單位很簡單。這意味著一個碗碟就是一個碗碟，一輛汽車就是一輛汽車。通常，它們不能被分解成更小的部分。雖然也有例外，比如一輛大卡車可以承載許多汽車；一輛大卡車可以被分解成一個大單位和許多通過道路的汽車。

在編程中，這要靈活得多。即使是一個插入或更新的 SQL 語句，也可以分解成更小的部分，更不用說下載任務、DFS 搜索或查詢了。

好的，現在我們已經考慮了處理單位。那麼，問題是我們應該一次處理多少個單位。它可以是一個到總單位數之間的任何數字。

這裡的問題是，一個工作的批次數量是固定的還是動態的。對於生成式 AI，輸入文本的總字符數是靈活的。它有一些上下文限制或輸入限制，但在其限制範圍內，它是靈活的。

使用洗碗機時，其內部空間有限。在該限制內，碗碟的數量是靈活的。我們通常會把需要清洗的碗碟盡可能多地放入機器。

對於程序，數據庫一次處理的 SQL 語句批次有限。在該限制內，它能處理的 SQL 語句數量是靈活的。但我們應該考慮從客戶端傳遞 SQL 語句到數據庫服務器的網絡任務，用戶能等待多長時間，以及如果批次中的一個單位任務失敗會發生什麼。

因此，對於我們應該一次處理多少個單位任務的問題，我們應該考慮工作的目標、下游消費者或處理者的限制，以及失敗的概率。

這種思考方式可以應用於許多事情。基本上有兩個問題需要考慮：什麼是單位任務，以及我們應該一次處理多少個單位。通過考慮這些問題，我們可能會得出一個最佳解決方案。