對本書的讚譽

Sev 的個人實踐經驗和策略可以為我的雇主省下數百萬美元。考量到這本書的價格和閱讀時間，這是一個相當不錯的投資回報。

— *Bar Shirtcliff*，軟體工程師

得益於 2010 年代中期的雲端資料技術演化，現今的資料工程師得以輕鬆地存取超大規模運算能力和巨量資料儲存設施，但這種巨大的變化也讓工程師必須對日常工作中的所有花費負責。這是我們一直在等待的一本書，它為監控、控制和高效能雲端資料系統的成本優化，提供了清晰、獨樹一格的見解。

— *Matthew Housley*
技術長及《*Fundamentals of Data Engineering*》的作者之一

眾所周知，現實世界的資料管道變化無常，並且隨著時間的推移和事物的變化，資料管道有可能會發生難以預料的問題。這本書是一個很好的資源，可以幫助您在代價高昂的資料管道出現問題之前，及早地介入處理。

— *Joe Reis*，《資料工程基礎》的作者之一

當談論在現今世界中設計和建立強大資料管道的方法時，本書可說是這幾十年來我見過最容易上手的指南。處理混亂且複雜問題時，往往沒有簡單的方法可以一次解決所有困難，此時，藉由本書提供的豐富背景資訊，並且參考書中以真實程式碼作為範例的詳細解說，它將成為全天候供您諮詢的專家。您將學會在成本、效能、開發時間、後續的長期支援、未來成長，以及構成當今複雜資料管道的眾多其他要素之間，取捨權衡。

— *Arnie Wernick*，資深技術專家、智慧財產權和策略方面顧問

前言

在我處理資料管道的經驗中，最昂貴的那次事件，是由一個程式臭蟲引起的：某個資料管道一直錯誤地轉換資料，直到好幾個月後我們的客戶注意到資料錯誤，才發現這個問題。

雖然許多問題都可能會導致這項結果。例如資料的變化太大而難以監控資料品質，加上由於測試資料已經嚴重過期，所以無法感知到資料品質有異。於是測試程式碼變更的唯一方法，是完整檢查資料管道，但這需要很長的時間且耗費巨大成本。除此之外，即使任誰都知道資料來源可能會發生不可預測的變化，但卻沒有在管道中進行資料驗證，偵測到究竟是何時發生變化。

為了解決這個錯誤，我們花費年度雲端費用預算中的一大部分，來重新計算錯誤的資料，更糟糕的是，這個問題也損害了客戶對我們的信任，甚至引發對該專案有效性的質疑。這是一份價值數百萬美元的合約，底下養著數十個工作職位，以提供協助近一億人的服務。這種規模的錯誤對每個肩負重責大任的資料工程師來說，都是他們的職業生涯中可能面臨的挑戰。事後回顧，我們應該要透過基於模式的驗證來捕捉這個錯誤，而這正是本書將讓您了解的內容。

當您在試圖控制雲端資料管道的成本時，通常會面臨一些權衡與取捨。例如：在滿足效能需求的前提，同時減少計算週期的浪費。利用可觀測性進行系統調試、除錯，在降低成本的同時，也不斷地改進原先的設計，並且注意在監控和日誌記錄上不要過度花費。提高測試覆蓋率，但需注意資料的不可預測變化，可能會使測試假設無效，且雲端服務

介面會引入額外的成本和複雜性。在不犧牲管道穩定性的前提下，使用低成本、可被中斷的計算執行個體。

本書整合了您需要迅速處理這類權衡與取捨的所需知識，並著重於介紹有效的監控、資料管道開發和測試，且特別針對雲端運算和儲存的設計，提供實務上的建議。一言以蔽之，本書將使您從一開始就為成功做好準備，並使您能夠以具備成本效益的方式，管理資料管道的演進。

我已經在批次和串流系統中使用了這些方法，這些解決方案涵蓋從只有幾千行的小量資料到 PB 級資料量的各種情況，包括明確定義的結構化資料和頻繁變化的半結構化資料。

本書目標讀者

本書內容所談及的主題涵蓋中階至進階的各個知識點，並且假設讀者對軟體開發的最佳實踐有一些了解，對雲端運算和儲存亦有基礎知識，以及對批次和串流資料管道的運作有簡略的概念。

我從開發資料管道的日常工作經驗中，將其濃縮提煉成本書內容。不管您已經是相關的從業人員，或是希望在未來加入這一行業的新手，都可以把這本書看作一個虛擬導師，它將提醒您關於建構資料管道時常見的陷阱，並且藉由各種資料管道專案範例，學習到扎實的行動方針。

如果您具備資料分析能力，您將在軟體方面找到最佳實踐的建議，幫助您建立可測試、可擴展的資料管道。這將有助於您串連資料分析需求、資料擷取需求和資料儲存需求，以建立完整的系統（end-to-end systems）。

開發速度和注重成本的設計是每個人都應該關注的領域，無論是工程師或擔任管理職。在這本書中，您將找到將品質融入開發過程、有效利用雲端資源和降低成本的建議；此外，還將了解監控的要素，不僅可以追蹤系統的健康度和執行效能，還可以洞察需要重新設計之處。

如果您管理一組資料工程團隊，也將找到有關高效開發的實踐祕訣、成本可能上升的領域，以及建立正確實踐的整體方法，以幫助您的團隊取得成功。

本書範例的故事背景

為了闡述「開發具成本效益的資料管道」的不同面向，本書使用一個虛構的社群網站範例，名為「蒼鷺點播」（Herons on Demand，HoD）。蒼鷺是世界各地都可以看到的雄偉猛禽，而 HoD 網站由因熱愛觀賞蒼鷺（*https://oreil.ly/Rdwwc*）而結下深厚友情的 Lou 和 Sylvia 所創立，作為經驗豐富的軟體開發者，他們正在獨立創業以追求自身夢想，那就是幫助他人以線上影片的方式，觀賞這些難以捉摸的生物，並推薦觀賞蒼鷺的最佳場域。

Lou 和 Sylvia 使用信用卡註冊了一個雲端服務供應商（CSP）的帳戶，立即獲得儲存、計算、資料庫和網站託管等服務；值得一提的是，如果這些基礎設施都由自己從零開始建置的話，會需要相當長一段時間才能達到這樣的成果。儘管網站一開始規模較小，但 Sylvia 和 Lou 的願望，是將有關蒼鷺的資訊傳遞給全球的鳥類愛好者，而這個需求使得雲端服務供應商（CSP）所提供的全球可用區域（availability zones）成為額外賣點。

雖然您可能會懷疑市場上是否有足夠需求，來支撐以蒼鷺為主題的公司，但事後結果顯示，這個領域的確存在一些非常值得開發的潛力。在「蒼鷺點播」（HoD）網站開張的幾個月後，網站上的造訪人數逐漸攀升，數百萬名使用者分享有關蒼鷺的資料。雖然 Lou 和 Syliva 在雲端中建構了整個 HoD 平台，因此平台本身可以根據需求快速擴展，不至於因流量過大而當機，但是雲端費用仍達到了他們的信用卡額度上限，幸運的是，一位對蒼鷺著迷的億萬富翁同意投資他們的公司，減輕他們的雲端服務成本支出壓力。

有鑑於「蒼鷺點播」的成功，它帶來一份龐大的蒼鷺資訊資料集，Lou、Sylvia 和他們的投資者考慮將這轉化為新的產品和服務，就在思考這個問題時，一所大學的鳥類學實驗室主動聯繫他們。該實驗室正在研究蒼鷺的遷徙，並認為「蒼鷺點播」的資料集可以幫助識別新興和瀕臨滅絕的遷徙地點。雖然 Lou 和 Syliva 對拯救蒼鷺的提案感到非常興奮，但那位億萬富翁卻不太看好，「幫助研究人員毫無利潤可言！」她說。

Lou 和 Sylvia 不願與現實妥協，於是提出了一個產品：蒼鷺識別服務（HIaaS）。這個服務將處理使用者上傳的資料，並在「蒼鷺點播」的資料庫中尋找匹配資訊，為使用者提供高可信度的蒼鷺識別。而該大學的鳥類學實驗室就是第一批使用者，是開始建立資料管道和測試這項服務的絕佳方式。

最終，那位億萬富翁同意了，但她希望看到一些投資回報。她對 Lou 和 Sylvia 提出條件，要求兩人以具成本效益的方式設計 HIaaS，以限制雲端費用和人員時間；在將這些約束情況考慮進來後，他們繼續與大學合作此計畫。

第 2 章

根據需求變化調整運算資源的規模

當我不寫書的時候，我喜歡在有空時去森林走走。而規劃這樣郊遊的一個重點在於弄清楚當天怎麼穿衣服，天氣預報、健行的困難程度以及我的步速，都是影響這個決定的因素。

洋蔥式穿法是解決這個難題的傳統招式。快步上山時，可以脫掉一些衣服以免流汗；停在山頂欣賞風景時，也可以重新穿上它們以保持溫暖。能夠隨意地增添、脫掉身上衣物，使我在野外依舊感到舒適和安全。

就像我可以穿脫衣物以適應不同情況一樣，您可以自定義資料管道的資源量，以應對不同的工作量。把這看作是一種動態權衡計算，這是一個迭代過程，涉及監控和調節任何可調整的事物。

第 1 章曾介紹**垂直擴展**的定義，它能改變資源容量；本章將專注於**水平擴展**，它能改變資源數量。正如「Design for Scaling in the Microsoft Azure Well-Architected Framework」中的描述（*https://oreil.ly/4xJU_*），增加更多資源會稱為 *scaling out*，而減少資源則稱為 *scaling in*。

不管是擴大管道能負荷的工作量，抑或是縮減資源規模，兩者都是關乎成本支出和整體效能的另一個重要主題。當然，您會希望藉由運算資源的擴展或縮減動作以節省成本支

出，但相信您也不希望拿資料管道的可靠性和效能作為代價交換。簡而言之，您希望擴展資源以應對大量資料工作量，但又不希望分配過多資源以免浪費，而造成不堪負荷的成本支出。

本章將重點介紹應用水平自動擴展機制至資料管道的部署方式，並說明要在何時、何地以及如何擴展資料工作量，帶您了解找到可應用擴展機制的機會、設計可自動擴展的資料管道架構，並導入水平自動擴展機制。本章最後的範例會整合所有內容，並概述您將在實際應用中看到的不同類型自動擴展服務。

找尋可應用擴展機制的機會

在探討擴展資料管道的方法之前，首先，必須先確保是否存在擴展資料管道的機會。擴展需要兩大因素：可變性和指標。

所謂的**可變性**，就是提供擴展的機會，缺此，就只能使用固定的運算資源。比方說，如果我在炎熱、陽光普照的日子裡健行，我就不會帶多餘的衣物，[1] 因為我知道不論這次健行多辛苦，或是走得多快，我都會覺得很熱，只有本章前言提及不同天氣條件，才會讓我要增減衣物。同理，在資料管道中，操作和工作量的各種變化，正是替您帶來擴展機會之處。

至於**指標**，則是用來識別需擴展時機的工具。處理大批資料時，記憶體的使用量會增加，此時指標會顯示需要更多的資源；而當處理大量資料的尖峰時間過去，資料量會逐漸地減少，同時 CPU 使用率也下降，也就提供了另一個縮減資源規模，以降低成本支出的機會。

資料管道中的多變性

可以從管道操作和資料工作量的角度，來思考資料管道中的多變性。**管道操作**是指管道處理一次資料的間隔時間，而**資料工作量**則涵蓋了資料量和資料的複雜程度。圖 2-1 說明管道操作和工作量都保持恆定的基本情況，在這個範例中，資料管道是一個每隔幾秒就監控溫度數值變化的串流系統，由於此情境中，一切都不會隨著時間而變化，因此沒有擴展的機會。

1 除非防曬也算一層衣物。

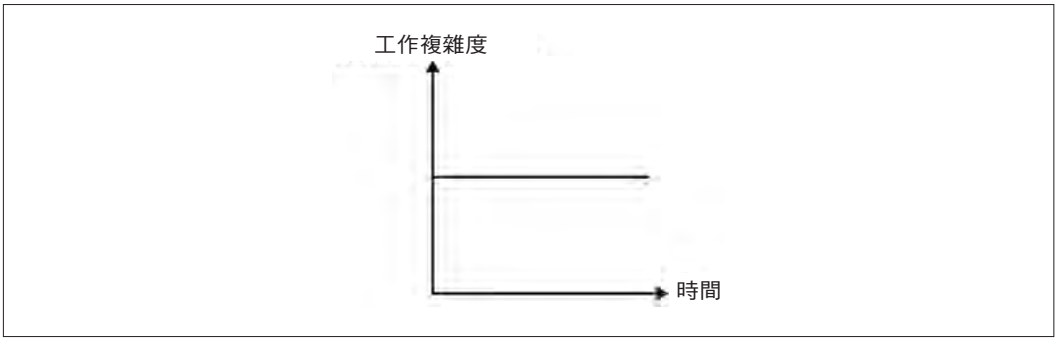

圖 2-1　固定的工作量、固定的操作

當資料工作量和管道操作開始出現變化時，就有更多機會擴展，以達到成本和效能的最佳化。如圖 2-2 所示，藉由操作的靈活與彈性，可以在只有少數工作正在執行時，抑或是資源完全閒置時，縮減資料管道的規模以節省成本支出。這類型作法包含批次處理資料的管道，以及間歇處理資料的串流管道。

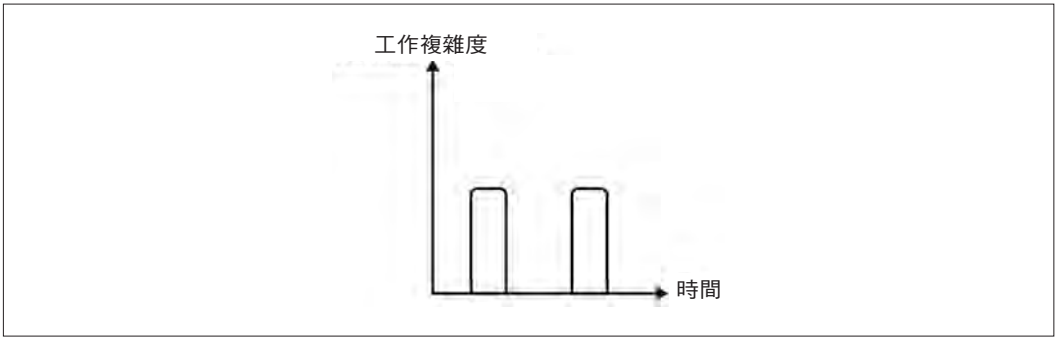

圖 2-2　固定的工作量、彈性的操作

在「不變的工作量、具備彈性操作」的情境下，需要注意的是，區分資源是基於擴展機制，或是因間歇資料處理作業而觸發。在第 1 章的批次資料處理管道中，無論是根據特定時間或是因外部事件，那些處理資料的作業都是受觸發而執行的。在另一個我曾參與過的資料管道例子中，由於該資料管道僅在特定時間區段執行資料處理作業，因此，該資料管道僅在該時段擴展運算資源，在其餘時間則會縮減運算資源至最低程度。這兩個例子看起來都如同圖 2-2 所示，但一個使用擴展機制，而另一個使用觸發來啟動所需的資源。

總之，資料管道中的管道操作與資料工作量的變化，凸顯了擴展的機會，至於如何確定擴展時機則取決於識別與這些可變情境相關的指標。關於這一點，可以透過以下問題，來評估一個資料管道的擴展能力：

- 管道操作如何隨著時間改變？
- 資料工作量會如何變化？
- 資料工作量或操作的變化將如何影響資源需求？
- 您怎麼知道資源需求正在改變？

來看看將此流程應用到一個資料管道的範例。

管道擴展範例

回想一下第 1 章的 HoD 批次處理資料管道，它每兩週處理一次鳥類調查資料，如下圖 2-5 所示：可以看見該管道每一次處理的資料量。

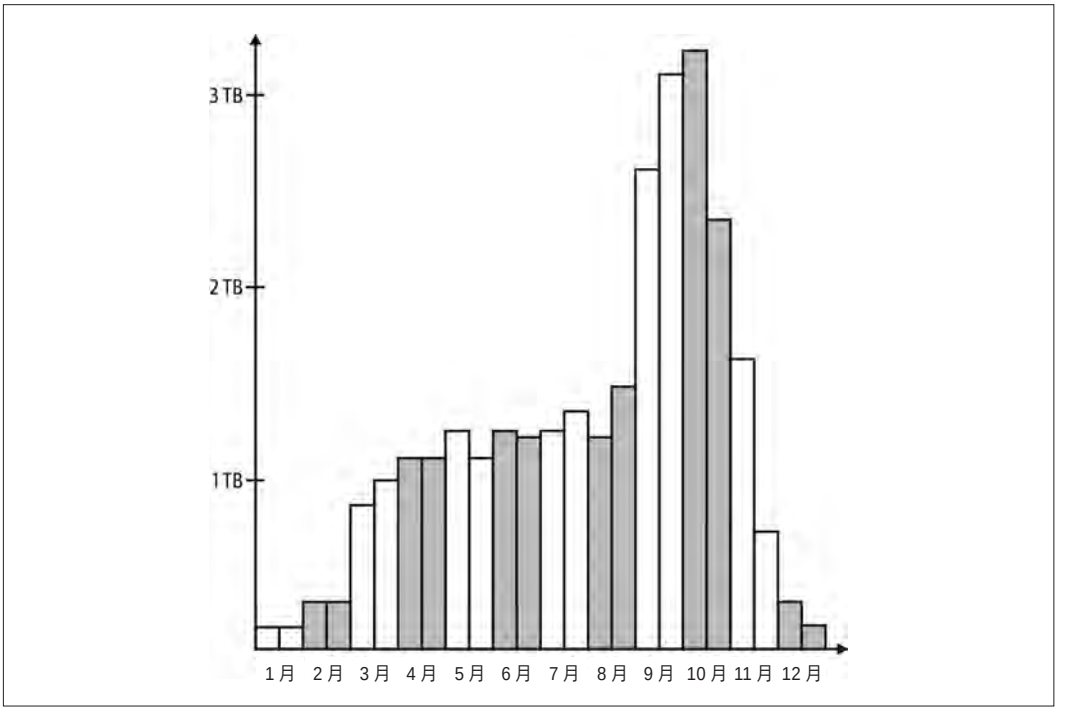

圖 2-5　兩週一次的資料工作量

現在來套用擴展性的那些問題。

管道操作如何隨著時間改變？執行資料管道工作的時機可以調整與改變，可以設定它每兩週只運行一次，或者依據需要而隨時運行。

資料工作量會如何變化？如柱狀圖顯示，資料量會隨著季節變化。

如圖 1-3 和圖 2-6 所示，鳥類調查資料並不是資料管道的唯一來源，該資料管道亦加入 HoD 網站中所上傳的社群資料，藉此使整體資料更為豐富。然而，來自社群媒體的資料量是不可預測的；它取決於目前正在處理的鳥類調查資料，以及 HoD 資料庫記錄該鳥類資料的方式。

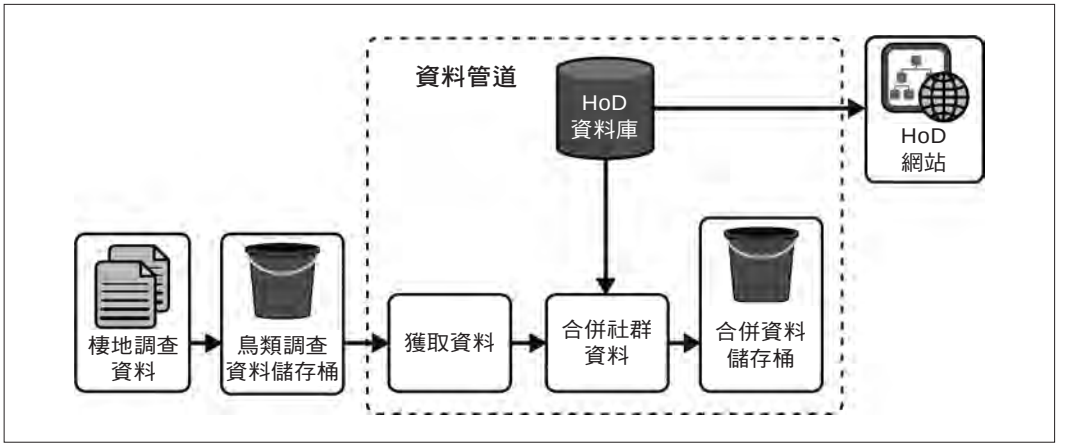

圖 2-6　HoD 批次處理資料管道

回到資料工作量會如何變化的問題上，首先，必須記住，鳥類調查資料和來自 HoD 社群上傳的資料，有著不同的資料量。

資料工作量或操作的變化將如何影響資源需求？根據第 1 章所見的管道效能測試結果，可知道實際的記憶體需求將取決於管道內的資料工作量。此外，別忘記這裡兩週才執行一次資料管道，因此，只有在執行工作的期間，才會需要部署運算資源。

鳥類調查資料和來自 HoD 社群上傳資料之間若有重疊，也會對運算資源需求造成影響。例如，若資料來源之間有明顯重疊，則當資料管道在執行「利用社群資料使鳥類調查資料更為豐富」的步驟時，由於這是一種合併資料的操作，或許會導致該步驟變成資

源密集型的步驟。換言之，若兩者資料來源之間並無重疊性，則該合併步驟可能僅需要少量的運算資源。

您怎麼知道資源需求正在改變？這是一個好問題。在回答它之前，讓我們先來看看當資料通過 HoD 資料管道時，資源需求會如何變化。

回想一下資料管道內部的變化，由於圖 2-6 的「使用社群資料，使鳥類調查資料更為豐富」步驟，屬於合併資料的操作，因此，它可能需要比「汲取資料」步驟更多的資源。有鑑於此，圖 2-7 說明一種可行方案的想法，即擴展運算資源，以執行「使用社群資料，使鳥類調查資料更為豐富」步驟，並在該操作完成後縮減資源規模。如圖所示，相較於白色框框的「汲取資料」步驟，以著色框框表示的「使用社群資料，使鳥類調查資料更為豐富」步驟，會需要更多資源。

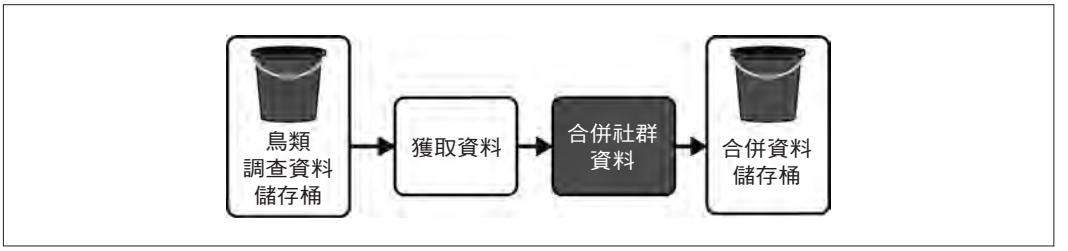

圖 2-7　擴展每個階段的資源。著色框表示有著更高的資源需求。

當鳥類調查資料與 HoD 社群所上傳的資料之間存在著您不知道的重疊，就會造成預期之外的資源需求，才能滿足「利用社群資料使鳥類調查資料變得更豐富」步驟。

圖 2-8 描繪這樣的情境：左側顯示鳥類調查資料與社群資料之間的關係，而右側則顯示「利用社群資料，使鳥類調查資料更為豐富」步驟的相應資源需求。若著色框框的顏色愈深，代表執行此合併操作，所需的資源就愈多。

總而言之，基於資料管道的不同處理階段與資料特性，所需的資源亦會有所變化。所以，**您怎麼知道資源需求正在改變？**

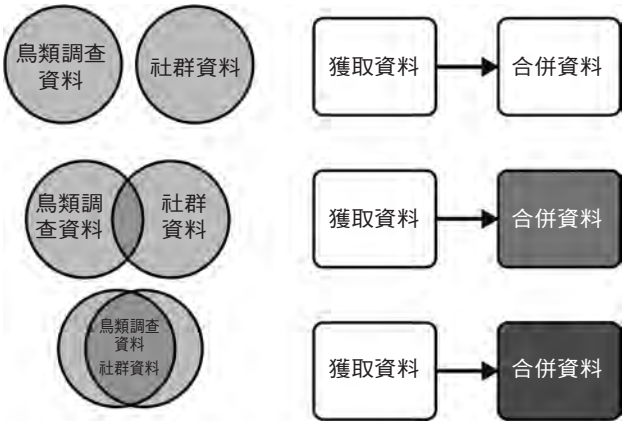

圖 2-8　跨資料管道作業的不同資源需求

根據第 1 章中對這個資料管道所做的效能測試，可知「利用社群資料，使鳥類調查資料更為豐富」對記憶體需求來說是很吃重的步驟。現在可進一步得知，隨著鳥類調查資料和社群資料之間重疊部分的增加，記憶體需求也會隨之上升。基於這個觀察，您可以將記憶體使用率持續上升作為指標，並用它來評估是否應擴展運算資源。相反地，當記憶體使用率持續下降時，或許可視為目前僅需要較低的運算資源，因此，也可考慮縮減資源規模，以降低成本支出。

透過擴展降低成本

在回顧擴展機制帶來的價值之前，請仔細地想想，如果您的資料管道使用固定的資源，其成本支出會是多少？如果想確保某種程度上的效能表現和可靠性，將需要為圖 2-8 中資料來源之間有大量重疊的情況，提供足夠的運算資源。如果您提供的資源不足，遇到有大量資料需要合併時，可能就會面臨運行時間過長，或管道作業失敗的風險。

而當資料來源之間幾乎沒有重疊的情況時，若採用固定資源的策略，將浪費大量運算成本支出。因此，可利用雲端服務的彈性，依據不同資料量，適當擴展或縮減運算資源的規模，從而在不犧牲處理較大工作量時，可能影響效能和可靠性的情況下，節省較小工作量的成本。

如圖 2-9 中所描述的擴展過程時間軸，圓圈表示運算資源，著色部分代表資源使用率，而每個運算資源下方的框框則代表工作量。時間點之間的差值並不等同於連續階段之間的相對延遲。

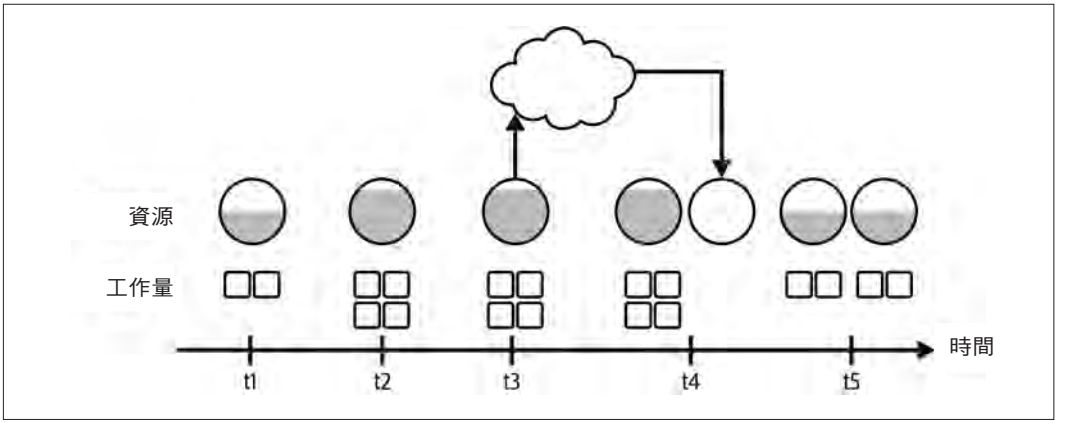

圖 2-9　執行擴展時，關於請求與重新分配資源的時間軸

從 t1 時刻開始，標準工作量使用了大約 50% 的運算能力。到了 t2 時刻，加倍的工作量帶來較高負荷，使用掉的運算能力逼近滿載；並且，增量的工作已超過監控指標的閥值。於是在 t3 時刻，系統觀察到該指標閥值逾界，並因此向平台請求更多的運算資源。

請回想第 1 章，當提出增加容量的請求後，增加額外的計算資源可能需要一些時間。接下來，在 t4 時刻，將額外容量加入至叢集，並在 t5 時刻，叢集將原本的工作量重新平衡至額外的計算能力上，至此，叢集擴展行動已經結束。最後，從 t5 時刻可知，當有額外運算能力來處理更多工作量時，資源使用率將降低回到 t1 時刻的水準。

上述流程在實務上的一個案例，是 Google 的雲服務（GCP）中，Google Cloud Compute（GCS）使用自動擴展機制來服務請求頻率極高的應用程式，可以在 GCP 的官方指南（*https://oreil.ly/8b89S*）中找到詳細說明。該文件指出，當 GCS 識別出請求頻率過高並試圖增加運算資源時，可能需要幾分鐘的時間。與此同時，由於過高的請求頻率已超出了叢集可用容量，響應可能會明顯減慢或失敗，為了避免這個問題，GCP 建議使用者逐步增加請求頻率，讓 GCS 有時間自動擴展機制，不要一下子要求增加太多請求頻率。

這點非常重要：**在 *GCS* 進行擴展叢集規模之前，如果發送的請求數量持續地增加，可能會有很高的機率發生故障**。除此之外，如果無法積極地擴展叢集規模以滿足大型工作量的需求，也會遇到這些同類型的效能和可靠性問題。

相反地，欲縮減叢集規模時，如圖 2-10，則操作順序將有些許不同。延續圖 2-9 中所描述的情境，在 t1 時刻，有兩個節點個別執行兩個工作單位，接著，到了 t2 時刻，其中一個節點減少一個工作單位，進而導致該節點的資源使用率下降。於是在 t3 時刻觸發監控指標的閥值，並因此向平台請求終止使用率過低的運算資源。

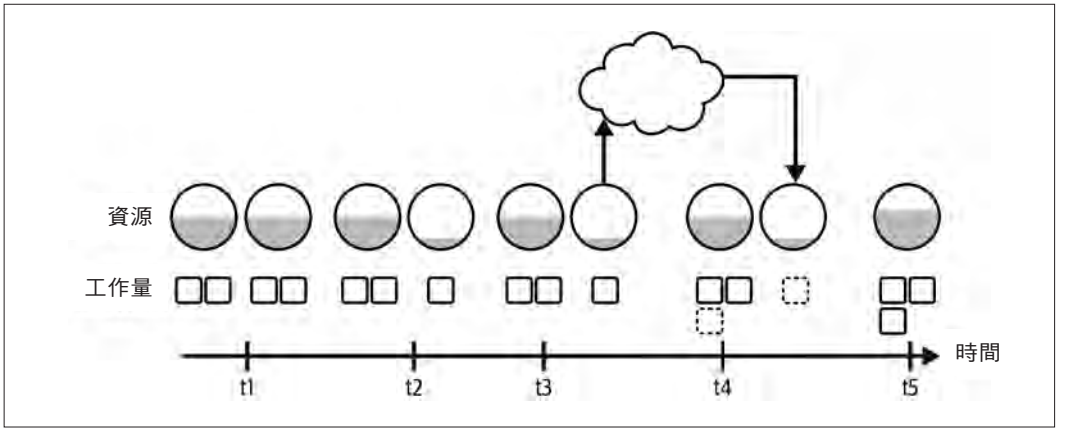

圖 2-10　縮減叢集規模時，移除資源和重新分配工作量的時間軸

如圖 2-10 所示，t4 時刻是一個很有趣的時間點。因為標示成虛線的工作單位可以有不同的行為表現。雖然在大多數的情境中，未完成的工作單元會移至其他可用節點上繼續執行，但在某些情況下，也可以等待該工作完成後才終止節點。如 Hadoop YARN 的功能之一：Graceful Decommission（*https://oreil.ly/iECdq*）就是一個很好的例子，它允許等待作業完成後才終止節點。

延續剛剛 t4 時刻的另一種討論情境：立即終止節點，並將未完成的工作單元移至其他可用節點上繼續執行。在這種情況下，其他節點必須重試原先執行程序做到一半的工作。例如，試圖擴展串流資料管道時就有可能會發生這種情況，但您不會希望使用如同 Hadoop YARN 的 graceful decommissioning 功能以作為縮減策略的一部分，因為這會卡住縮減叢集規模的進行。此時，可以試著使用先前提及的檢查點（checkpointing）機制，來執行這個重試工作。

如圖 2-10 所示，縮減叢集規模完成之後仍然會保留一個工作節點，如有必要，該工作節點會從移除節點中取得原本的工作量，並繼之執行。

本章總結

如同生活中的多樣性可為生命增添不同的酸甜苦辣，資料管道與資料量的變化也是一項指標，用以顯示不僅可以利用擴展機制使資料管道發揮其預期作用，亦可利用雲端服務的彈性來縮減內部成本支出，滿足日益漸增的需求。

回答以下問題，可以幫助您找出資料管道中的變異性來源及其預測因子：

- 管道操作如何隨著時間改變？
- 資料工作量會如何變化？
- 資料工作量或操作的變化將如何影響資源需求？
- 您怎麼知道資源需求正在改變？

如果可改變資料管道的操作以因應工作量比較少的情境，那即可藉此措施降低成本支出。資料量的變化也可帶來降低成本支出的機會，例如，當資料量較小時，可縮減運算資源規模，反之，在處理較大資料量時，則新增運算資源以滿足效能需求。

您可以在多個層次上擴展。例如在系統層面上，當資料管道具備週期性操作的特質時，可以利用事先定義的自動擴展規則，在資源利用率相對低點的時候降低成本支出。當資料管道的操作變得較不可預測時，可以使用資源使用率和系統指標來觸發擴展叢集規模。

至於在資料處理的層面上，工作量的變化不僅可為縮減叢集規模以因應簡單的工作量，同時亦支援擴展叢集節點，以支撐更高的容量與更複雜的處理情境。除此之外，您還可以利用 Spark 的動態資源分配來擴展資料處理工作，以充分因應橫跨各個資料處理階段的不同資源需求。

若想知道適合執行擴展的時機，首先，需要識別相關且有意義的指標，以及從這些指標數值中，是否可以得知目前叢集需要增加或減少資源。對不同工作量進行效能測試，可以幫助您確定這些指標的原始表現以作為評估基準點，以及為後續擴展決策提供適當的閥值和觀察區間。

水平擴展的基本要素之一，就是資料管道的設計須能充分利用分散式資料處理技術。這包含深思熟慮的程式碼設計、資料分區規劃，以及對節點間資料交換（shuffle）的了解，這些知識都將幫助您充分利用擴展機制。