
推薦序

在不斷發展的雲端原生技術領域中，觀察應用程式的性能和健康狀態已不再是一種奢侈，而是必要的關鍵。隨著微服務架構成為常態，分散式系統擴展，以及資料量的爆炸性增長，傳統的監控工具難以跟上腳步。這就是 OpenTelemetry 崛起以改變遊戲規則的時刻，它提供了一種標準化且供應商中立的可觀測性方法。OpenTelemetry 不僅僅是一種技術；它還代表了一種典範轉移，從僅僅監控過渡到完整的可觀測性。OpenTelemetry 正在將行業從孤島式運作轉變為統一的遙測。

正如作者 Ted Young 和 Austin Parker 所解釋的，OpenTelemetry 是關於採用統一的遙測資料驅動方法來進行可觀測性，利用開放標準如 OpenTelemetry 協議（OTLP），並賦予建造和操作完全可觀測、高度韌性、高性能的雲端原生應用程式的能力。

《OpenTelemetry 學習手冊》可作為你解鎖 OpenTelemetry 力量的全面指南。無論你是一位在應對分散式追蹤複雜性方面經驗豐富的工程師，或試圖理解基礎知識的新手，還是一家欲開始可觀測性之旅的組織，這本書都將為你提供知識和實用洞見，幫助你導航這項變革性技術。

作者強調，可觀測性需要理解雲端原生思維更廣泛的背景和固有的挑戰。例如，微服務架構雖然提供了敏捷性和可擴展性，卻引入了新的複雜性。為單體應用程式設計的傳統監控工具，往往難以捕捉服務之間錯綜複雜的互動和依賴關係。這種缺乏連貫的可見性導致了可見性缺口，使得難以精確定位性能瓶頸、診斷問題以及確保應用程式的健康。

本書能突顯 OpenTelemetry 透過提供統一且廠商中立的方法，來蒐集和導出遙測資料之辦法，直接面對這些挑戰。這種統一的方法使用指標、追蹤、日誌和性能剖析，為你的應用程式的健康和性能提供了一個相關聯的視圖。

作者深入探討了 OpenTelemetry 的細節，引導我們了解 OpenTelemetry 的核心概念，並追求不同程式語言和框架（如共享套件和共享服務）的檢測策略。他們闡明了使用 OpenTelemetry 蒐集器蒐集和處理遙測資料的最佳實踐；他們調查了為 Kubernetes、無伺服器函式和資料流平台等擴展遙測蒐集的部署模式。他們將向你展示如何透過平衡廣泛方法與深入方法、集中式架構與去中心化架構等，來建造可擴展的遙測流水線。最後一章探討了進階話題，如生成式 AI、FinOps 和雲端永續性。

這個是令人興奮的時代。隨著雲端原生服務和 AI 應用的世界匯聚，使用遙測資料來了解大規模模型行為至關重要。這也是 OpenTelemetry 下一個重大進展將會是提供開放框架的原因，以完全支持智能、分散式 GenAI 應用的可觀測性。作為一種實踐，可觀測性必須納入可行的 AI 模型來蒐集和分析大規模遙測資料。

所以，打開這本書，深入 OpenTelemetry 的世界，並為你的雲端原生旅程解鎖可觀測性的力量。記住，通往精通的道路始於第一步，這本書是你在那次旅程中的第一步和接下來步驟的指南。

享受這段學習之旅！

— Alolita Sharma
Palo Alto, California
February 2024

Alolita Sharma 是 OpenTelemetry 治理委員會的成員，並且已經為 OpenTelemetry 項目貢獻了五年以上。她是 CNCF 可觀測性技術顧問組（TAG）的共同主席，並領導蘋果公司的 AIML 可觀測性實踐。她在 OpenTelemetry、可觀測性 TAG、Unicode 和 W3C 等領域為開源和開放標準做出了貢獻。Alolita 在 AWS 的可觀測性、基礎設施和搜索工程領域也展現了強大的領導力，並且在 IBM、PayPal、Twitter 和維基百科管理過工程團隊。

前言

在過去的十年中，可觀測性已從一個在 Monitorama 或 Velocity（已停辦）等活動中討論的小眾學科，成長為一個價值數十億美元的行業，影響著雲端原生世界的每一個角落。然而，有效的可觀測性關鍵在於高質量的遙測資料。OpenTelemetry 是一個旨在提供這種資料的項目，在此過程中，開啟了可觀測性工具和實踐的下一代。

如果你正在讀這本書，很有可能你是一位可觀測性實踐者，也許是一位開發者或一位 SRE，你對於如何在營運環境中剖析和理解複雜系統感興趣。你會拿起這本書，可能是因為你對 OpenTelemetry、它的結合方法，以及它與歷史監控框架不同之處感到興趣；也或許你只是想了解這股風潮所為何來。畢竟，在短短五年內，OpenTelemetry 已從一個想法成長為世界上最受歡迎的開源項目之一。無論你為何而來，我們都很高興見到你。

寫這本書的目標不是創造一本 OpenTelemetry 的「缺失手冊」，你可以找到許多文件和教學，以及幾本深入探討在特定語言中實施 OpenTelemetry 的優秀書籍（詳情請見附錄 B）。我們的目標是提供一本全面的學習 OpenTelemetry 的指南，希望你不僅了解不同部分，而且還了解它們結合在一起的方法和原因。這本書應該為你提供基礎知識，不僅需要在營運系統中實施 OpenTelemetry，還能夠擴展 OpenTelemetry 本身；無論是作為項目的貢獻者，還是將其作為組織可觀測性策略的一部分。

總的來說，這本書主要分為兩部分。在第 1 章至第 4 章中，會討論監控和可觀測性的現狀，並向你展示了 OpenTelemetry 背後的動機。這些章節幫助你理解支撐整個項目的基礎概念。它們對於首次閱讀者來說不僅價值連城，對於已經實踐可觀測性一段時間的人也同樣寶貴。第 5 章至第 9 章則進入具體的使用案例和實施策略，以之前章節介紹的概念為主，討論其背後的運作原理，並提供指引，好讓你在各種應用和情景中實際實施 OpenTelemetry。

現代可觀測性的現狀

歷史不是過去，而是從特定視角繪製的過去地圖，用來幫助現代旅人。

— Henry Glassie，美國歷史學家¹

這是一本關於大規模分散式計算機系統固有難題的書，以及如何應用 OpenTelemetry 來幫助解決這些問題。

現代軟體工程著迷於使用者體驗，而使用者要求極速的性能。調查顯示，如果電商網站加載時間超過 2 秒，使用者就會離開這個網頁（<https://oreil.ly/tZ9tY>）。你可能已經花了相當多的時間嘗試優化和排查應用程式性能問題，如果你和我們一樣，會因為這個過程的不優雅和低效感到沮喪，資料不是不夠就是太多，而且現有的資料可能充滿了不一致性或測量不清晰的問題。

工程師還面臨著嚴格的系統正常運行時間要求。這意味著在問題導致系統崩潰之前就要識別並減輕問題，而不僅僅是等待系統故障。這也意味著需要從初步診斷快速過渡到問題解決。為此，你需要資料。

但你需要的不僅僅是任何資料；你需要相關聯的資料，已經組織好的、準備好由計算機系統分析的資料。正如你將看到的，這種層次的組織資料並不容易獲得。實際上，隨著系統規模的擴大和變得更加異質化，找到需要分析問題的資料變得更加困難。如果以前像在乾草堆中尋找針，現在更像在一堆針中尋找針。

¹ Henry Glassie，《Passing the Time in Ballymenone: Culture and History of an Ulster Community》（費城：賓夕法尼亞大學出版社，1982 年）。

OpenTelemetry 解決了這個問題。透過將單獨的日誌、指標和追蹤轉化為一個連貫的、統一的資訊圖表，OpenTelemetry 為下一代可觀測性工具奠定了基礎。而且由於軟體行業已經廣泛採用 OpenTelemetry，這一代的工具就建立於我們撰寫本書內容時。

時代正在改變

技術如潮水般湧來。正如我們在 2024 年撰寫這段內容時，可觀測性領域正經歷至少 30 年來的首次真正海嘯。你選擇了一個好時機來閱讀這本書並獲得新視角！

雲端計算和雲端原生應用系統的出現，導致了建造和操作複雜軟體系統實踐的巨大變革。然而，沒有改變的是，軟體運行在電腦系統上，你需要了解這些電腦系統在做什麼，才能理解你的軟體。儘管雲端計算試圖將計算的基本單元抽象化，但二進制仍然在使用位元和位元組。

不論你運行程式在多區域的 Kubernetes 叢集還是在個人電腦上，都會有一樣的問題：

「為什麼它這麼慢？」

「是什麼占用了這麼多的 RAM？」

「這個問題是什麼時候開始的？」

「問題的根本原因在哪裡？」

「我該如何解決這個問題？」

天文學家和科學傳播者 Carl Sagan 曾說過：「了解現在必須知道過去。²」這在這裡確實適用：要了解為什麼一種新的可觀測性方法如此重要，首先需要熟悉傳統的可觀測性架構及其局限性。

這可能看起來像是對基本資訊的重述！但可觀測性混亂已經存在了很長一段時間，以至於大多數人已經形成了相當多的先入為主觀念。因此，即使你是專家——尤其是如果你是專家——擁有新的觀點也是很重要的。讓我們開始這段旅程，先來定義本書將使用的幾個關鍵術語。

² 「Cosmos: A Personal Voyage」第 1 季第 2 集「One Voice in the Cosmic Fugue」，由 Carl E. Sagan 製作主持，Adrian Malone 執導（Arlington, VA：公共廣播電視，1980 年）。

可觀測性：需要了解的關鍵術語

首先，可觀測性觀測的是什麼？就本書的目的而言，我們觀測的是分散式系統。分散式系統是指其元件位於不同的網路電腦系統上，透過相互傳遞訊息來通訊和協調其行動的系統³。有許多種類的電腦系統，但我們關注的是這些系統。



什麼是分散式系統？

分散式系統不僅僅是在雲端運行的應用程式、微服務或 Kubernetes 應用程式。採用服務導向架構的巨集服務或「單體式服務」、與後端通訊的客戶端應用程式，以及移動端應用程式和網頁應用程式，都可以算是某種程度上的分散式系統，並從可觀測性中受益。

在最高層次上，分散式系統由資源和請求組成：

資源（Resource）

這些是構成系統的所有物理和邏輯元件。物理元件，如伺服器、容器、處理程序、RAM、CPU 和網路卡，都是資源。邏輯元件，如客戶端、應用程式、API 端點、資料庫和負載平衡器，也是資源。簡而言之，資源是構成系統的一切。

請求處理（Transaction）^{譯註}

這些是指示系統利用和協調所需資源來代表使用者執行工作的請求。通常，這些請求處理由一位真實的人啟動，他正在等待處理完成。訂購機票、叫車共乘和加載網頁都是此類請求處理的例子。

我們如何觀察這些分散式系統？除非它們發出遙測，否則沒辦法。遙測是描述系統工作內容的資料，沒有遙測，系統就只是一個充滿神祕感的巨大黑盒子。

許多開發人員對於遙測這個詞感到困惑。這個詞有很多不同的意思，在本書以及系統監控中，我們所做的區分是使用者遙測和性能遙測：

3 Andrew S. Tanenbaum 和 Maarten van Steen，「分散式系統：原理與範例」（Upper Saddle River, NJ: Prentice Hall, 2002）。

譯註 這裡翻譯成請求處理，書本原文是 *Transactions*，如果在資料庫領域翻譯成事務處理是適當的，用來描述確保資料一致性和完整性的操作。然而，在這裡描述的是使用者級的互動操作，這些操作是由用戶發起並等待系統回應的。這些操作不僅僅涉及資料庫的事務處理，還包括系統內部資源的協調和使用。

使用者遙測

指的是有關使用者透過客戶端與系統互動的資料：按鈕點擊、網路會話持續時間、有關客戶端主機的資訊等等。你可以使用這些資料來了解使用者如何與電子商務網站互動，或者訪問網頁應用程式的瀏覽器版本分布情況。

性能遙測

這不主要用於分析使用者行為；相反地，它提供給維運人員有關系統元件行為和性能的統計資訊。性能資料可以來自分散式系統中的不同來源，並為開發人員提供了一條「麵包屑路徑」，將因果關係連接起來。

用更直接的語言來說，使用者遙測資料會告訴你某人將其滑鼠游標懸停在電子商務應用程式中的結帳按鈕時間。性能遙測資料則會先告訴你載入該結帳按鈕花了多少時間，系統在這個過程中使用了哪些程式和資源。

使用者和性能遙測資料之下包含不同類型的訊號，訊號是一種特定形式的遙測資料。事件日誌是一種訊號，系統指標是另一種訊號，持續性能分析又是另一種，這些訊號類型各自有不同的用途，它們並不真正可以互換，你無法透過查看系統指標，就推導出構成使用者互動的所有事件，也無法只透過查看請求處理日誌來推導出系統負載，需要多種訊號才能深入了解整個系統。

每個訊號由兩部分組成：檢測工具（**Instrumentation**）：程式內部發出遙測資料的程式碼，以及將資料發送到網路上的分析工具的傳輸系統，會在那裡實際觀測。

這強調了一個重要的區別：將遙測和分析混為一談很常見，但要理解發送資料的系統和分析資料的系統是相互獨立的。遙測是資料本身，分析則是處理資料。

最後，遙測加上分析等於可觀測性。了解如何將這兩個部分結合成一個有用的可觀測性系統就是本書的重點。

可觀測性是一種實踐

可觀察性不僅僅停留在遙測和分析上；它是一種組織實踐，類似於 DevOps。在許多方面，可觀測性是現代軟體開發實踐的基礎，它支撐著我們所做的一切，從持續整合和部署到混沌工程、開發人員生產力等等。你的可觀測性來源像你的團隊和軟體一樣廣泛多樣，可以蒐集、分析這些資料，並用於持續改進整個組織。我們希望藉由本書，你能夠掌握建立一個立基於 OpenTelemetry 的組織內可觀測性實踐所需的基礎知識！

本書將引導你學習 OpenTelemetry。它不是為了要取代 OpenTelemetry 文件，該文件可以在項目的官方網站上找到（<https://opentelemetry.io>）。相反地，本書解釋了 OpenTelemetry 的哲學和設計，並提供了如何有效運用它的實用指南。

在第 2 章，我們將解釋 OpenTelemetry 帶來的價值主張，以及你的組織如何從使用基於開放標準的檢測工具替換專有檢測工具中受益。

在第 3 章，我們將深入探討 OpenTelemetry 模型，並討論追蹤、指標和日誌等主要的可觀測性訊號，以及它們是如何透過上下文相連。

在第 4 章，我們將在 OpenTelemetry 演示中實際操演 OpenTelemetry，為你提供其元件的概覽以及 OpenTelemetry 如何融入可觀測性堆疊。

在第 5 章，我們深入應用程式的檢測，並提供一個清單以幫助確保一切正常運作並且高品質遙測。

在第 6 章，我們討論如何檢測 OSS 函式庫和服務，並解釋為什麼函式庫維護者應該關心可觀測性。

在第 7 章，我們回顧了監測軟體基礎設施的選項：雲端服務提供商、平台和資料服務。

在第 8 章，我們詳細討論了如何以及為什麼使用 OpenTelemetry 蒐集器建造不同類型的可觀測性流水線。

在第 9 章，我們提供了如何在你的組織中部署 OpenTelemetry 的建議。由於遙測是一個跨團隊問題，對追蹤來說尤其如此，所以在推出新的可觀測性系統時存在組織上的陷阱。本章將提供確保成功推出的策略和建議。

最後，我們的附錄包括有關 OpenTelemetry 項目本身結構的有用資源，以及進一步閱讀和其他標題的連結。

如果你是 OpenTelemetry 的新手，我們強烈建議你仔細閱讀第 4 章。之後的章節可以按任何順序閱讀，隨意跳到對你當前任務最相關的部分。

人們需要什麼？

遙測和可觀測性擁有許多利益相關者。不同的團隊和個人對於可觀測性系統會有不同的需求，很自然地，他們對遙測資料本身也會有不同的要求。OpenTelemetry 如何滿足這些廣泛且常常相互競爭的利益？

在本章節中，我們將討論 OpenTelemetry 為開發者和維運人員以及團隊和組織帶來的好處。

開發者與維運人員

建立和維運軟體的人員需要他們的可觀測性資料具有高品質、高度上下文內容、高度相關性和分層等特點。他們需要遙測是內建的，而不是稍後必須添加的東西。他們需要它能夠在許多來源中持續普遍存在，並且需要能夠以一致的方式修改它。無論是改變內建的遙測還是添加新的遙測，跨多種程式語言、運行環境、雲端平台等。

如今，開發者使用檢測工具函式庫來建立這些遙測資料，他們有許多現有選擇，例如 Log4j、StatsD、Prometheus 和 Zipkin。專有工具還提供了自己的檢測功能 API 和軟體開發套件（SDK），以及對流行框架、函式庫、資料庫等的內建整合。

最終，這些檢測工具函式庫和格式對開發者和維運人員來說非常重要，因為它們定義了你能如何，以及如何有效地，透過遙測來模擬你的系統。檢測工具函式庫的選擇可能限制了你的系統的有效可觀測性：如果你不能發出正確的訊號，帶有正確的上下文和語意，你可能會發現自己根本無法回答某些問題。開發者在學習可觀測性時面臨的最大挑戰之一，是每個人做事的方式都略有不同。擁有強大、集中化的平台工程和內部工具團隊的組織，可能提供強大且整合良好的檢測工具函式庫和遙測能力，但許多組織並非如此。

可觀測性的一個推動問題是，系統過於龐大和複雜，而難以完全掌握並理解它們。當然，一直以來都存在於大型且複雜的軟體系統中，但現在真正不同的是變化的速度和由此導致的人類理解的喪失。在一個節奏較慢的世界裡，曾經有人理解應用的細微差別以及它們是如何組合在一起的，我們稱之為質量保證（QA）。隨著時間的推移，隨著越來越多組織拋棄傳統的 QA 流程，並用持續整合和交付來替代，人們越來越難以吸收系統的「形狀」。我們越是加快速度，就越需要無處不在、高質量的遙測資料，來描述發生的事情及原因。

除了檢測工具函式庫和高質量的遙測資料之外，維運人員還需要一個豐富的工具生態系統來幫助他們蒐集和處理遙測資料。當你每天產生 PB 級大小的日誌、指標和追蹤時，你需要一種方法來切割噪音，以便找到訊號，這不是一件小事！簡單地說，有太多的資料需要儲存和稍後分析查看，而且大部分資料可能並不十分有趣。因此，維運人員依賴能夠產生各種訊號的檢測工具，以及幫助他們過濾不重要事物的工具，以便他們能夠與開發者一起確保系統的可靠性和韌性。

團隊和組織

可觀測性不僅僅是為開發者設計的。其利益相關者還包括安全分析師、項目經理和高階管理層。他們可能需要對相同資料的不同視角，以不同的解析度來查看，但可觀測性仍然是任何組織威脅態勢、業務規劃和整體健康的關鍵部分。

你可以將這些視為「業務」的需求，但它們不僅僅如此。每個人都能從以下方面受益：

- 開放標準，避免供應商鎖定
- 標準資料格式和傳輸協議
- 可組合、可擴展且紀錄完善的檢測工具函式庫和對應的工具

可預測性對大多數組織來說極具吸引力，這就是為什麼組織中的標準化流程與程序存在的原因！他們以效率換取降低風險（大多數企業最不喜歡的字）。採用創新實踐冒險沒問題，但在不確定你的應用是否正常運行和維運時，冒險就不是個好主意了。因此，標準是組織及其可觀測性需求的當務之急。

基於標準的方法有許多好處。可維護性（Maintainability）就是一個例子。採用開放格式意味著開發者和團隊有更多的培訓機會。與其讓新工程師慢慢適應你的定制內部解決方案，不如讓他們學習如何使用開放標準檢測，並將這些知識帶入工作。這不僅提高了你維護現有檢測工具的能力，也有助於快速讓新開發者成為團隊的高效成員。

開放標準不僅風險較低；它們還具有未來性。環顧四周，從 2021 年到 2023 年，行業見證了多輪整合、收購以及大大小小的可觀測性產品失敗。在過去的 20 年中，我們見證了多種指標格式的建立和流行，但最終都或多或少地由新進者取代或降低了地位。

開放標準和開源不只是「可有可無」，它們在評估和建立你的可觀測性實踐時至關重要。你不需要看得太遠就能看到完全依賴專有解決方案的一些缺點。Coinbase（<https://oreil.ly/606GK>）2022 年在 Datadog 上花費了 6500 萬美元！我們並不是說這不值得，但天哪，這真是一大筆錢。

對組織來說最後一個重要因素是兼容性。你不太可能拔掉你現有的（功能性）檢測工具，只為了換成新的東西，而在大多數情況下這樣做是不明智的，除非你能獲得顯著更多的價值。關於這一點並沒有許多硬性規定，所以你需要的是能夠搭建舊的和新的橋梁，採納新的實踐同時保持你已有的東西，並將現有的遙測資料提升到標準格式。

為什麼使用 OpenTelemetry？

鑑於這些眾多利益相關者的所有需求，OpenTelemetry 為何會成為理想的解決方案？在最高層面上，OpenTelemetry 提供了兩個基本價值，這在其他地方找不到。

通用標準

OpenTelemetry 解決了當前可觀測性狀況中固有的問題。它提供了一種建立高質量、無所不在的遙測資料方法。它提供了一種標準方式來表示和傳輸遙測資料到任何可觀測性前端，消除由供應商所鎖定的標準。它旨在使遙測成為雲端原生軟體的內建功能，並且在許多方面它正在實現這一目標。截至本書撰寫時，三大主要雲端提供商：亞馬遜、Azure 和 Google Cloud Platform，都支持 OpenTelemetry，並正朝著將它標準化的方向前進。所有主要的可觀測性平台和工具都以某種方式接受 OpenTelemetry 資料。越來越多的函式庫和框架每個月都在採用 OpenTelemetry。

OpenTelemetry 預示了一個遙測會真正成為商品的未來，並致力於實現這樣的未來。它建造的未來是所有軟體都在表面之下建立豐富的遙測資料流，你可以根據自己的可觀測性目標挖掘和選擇所需資料。它不僅僅是一個新興標準，在這一點上，它是不可避免的，也是你需要採用的東西。

相關資料

OpenTelemetry 不僅僅是對以往實踐的規範化。為了推動該領域向前發展，下一代可觀測性工具需要有效地模擬維運平台在調查其系統時執行的工作流程。它們還需要運用機器學習來揭示那些可能難以直觀理解的相關性。

只有當所有遙測資料都規範化並相互關聯時，才能實現流暢的工作流程和高質量的相關性。OpenTelemetry 不僅僅是在同一地點堆疊的追蹤、指標和日誌。所有這些部分都是同一資料結構的一部分，連接成一個單一的圖表，描述了整個系統隨時間的變化。