

運算型儲存讓運算更接近數據

Neil Werdmuller 2020 年 9 月 3 日

數據產生的數量持續爆炸性成長。根據分析公司 IDC 的資料，數據產生、擷取或複製的數量，預計將由 2018 年的 33 ZB(zettabytes) 增加至 2025 年的 175 ZB。為了從這些數據實現價值，我們必須把它處理轉換成有意義的洞察。我們越來越清楚我們無法持續擴充以運算為中心的架構規模，焦點需放在從龐大數量的數據所在地產生洞察，也就是儲存裝置裡。這種情況正帶動以數據為中心的運算型儲存的快速發展。

今日越來越多的數據儲存在儲存碟中，但儲存這些數據與處理它的運算端，幾乎總在不同之處。在儲存端與運算端之間移動大量數據的方法無法擴充規模(儲存碟今日通常為 16TB，且容量持續增加中)，這使得從數據擷取可轉換成具附加價值且同時讓服務組織受益的洞察變得很難。

在傳統的儲存模型中，數據儲存於傳統硬碟(HDD)與固態硬碟(SSD)中，並由通常是伺服器的外部運算裝置進行存取與轉移。運算型儲存把數據處理擺在數據儲存所在的儲存碟裡，讓洞察與價值直接從數據產生。

運算型儲存是什麼？為何重要？

運算型儲存的重點，就是讓儲存裝置變得更聰明以直接從數據儲存的地方處理數據。這種方法降低把大量數據移至外部進行處理，並且可以帶來許多的好處，例如降低延遲與頻寬的使用，提升安全性與節省能源。換句話說，數據作業負載直接在儲存控制器本身進行處理。

對於應對許多機器學習(ML)或分析應用，以及從物聯網到終端運算等其它使用場景的即時處理需求，應用運算型儲存極為關鍵。以物聯網場景為例，部署數量的加速將產生大量原始、無結構數據，而這些數據會移至伺服器進行儲存與處理。不過，並非所有被擷取的數據都是相關的。

我們舉一個例子：一座大型停車場內的監視攝影機系統會錄下汽車的車牌號碼以及車輛進場與出場的時間，以便依停車時間計費，同時也達到安全的目的。這裡我們關注的資訊是車牌，而不管車輛進出場，把所有的大型影像或影片串流檔移至伺服器進行影像處理，是相當沒效率的。

有了運算型儲存，每個攝影機都會把資料串流到本地的儲存碟，然後由運算單元直接在儲存碟上辨識汽車的車牌。直接在儲存碟上執行 ML 與影像辨識，並且只把取自數據的洞察(車牌號碼與時間)傳回給伺服器，這是非常有效率的。此外，當一座停車場內有多台攝影機，而每台攝影機都配有一個儲存碟，那麼更多的攝影機就會有更多的儲存碟，我們也會在適合的地方擁有更高的運算力。這會讓系統更有效率，也更容易擴充規模。



我們還可以舉出其它許多運算型儲存可以帶來顯著影響的使用場景。常見的例子包括：

- 資料庫加速：直接對數據執行運算
- 卸載：直接對數據進行壓縮/加密/編碼/重複數據刪除/等
- 內容遞送網路(CDN)：很容易可以促成極為本地化的內容遞送
- 人工智慧/ML：從大量的數據直接產生洞察
- 邊緣運算：運行 Linux 的運算型儲存碟(CSD)，是自足式的小型伺服器。
- 影像分類：可以在儲存數據的地方直接對數據進行元資料標籤
- 影片：對大型檔案在地進行運算以產生洞察
- 交通運輸：直接處理儲存在車輛上的遙測數據

反觀今日傳統的儲存碟，把數據一路由裝置搬到伺服器來處理，這樣會造成：

- 需要額外的時間、能源、頻寬與延遲
- 增加資訊移動過程中未授權存取的可能性

倘若系統中連接到伺服器的回程網路只提供有限的頻寬，或是相當的昂貴，此時運算型儲存的好處包括可以大幅降低整體擁有成本(TCO)。額外的好處包括：

更快的回應時間與降低延遲

把智能移至需要的地方，可以讓我們以近乎即時的速度達成結果。數據不用封裝在協定中，然後透過路由器與交換機移動與複製，最後在伺服器端還原才能進行處理。

降低能源消耗

不會再有需要耗電並且會產生熱量的大量數據轉移

安全與隱私

數據從頭到尾並未離開儲存碟，只有洞察才會送回，可以降低資料外洩的風險

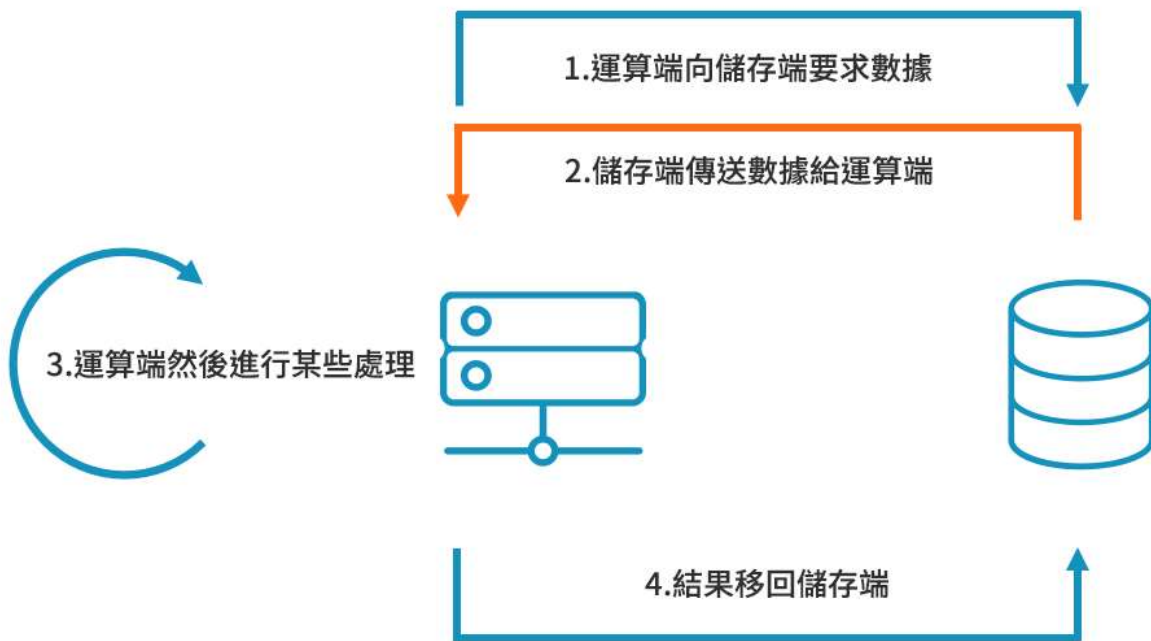
擴充性

由於運算在儲存碟內進行，增加更多的儲存碟，意味在數據儲存的地方增加更多的運算力。

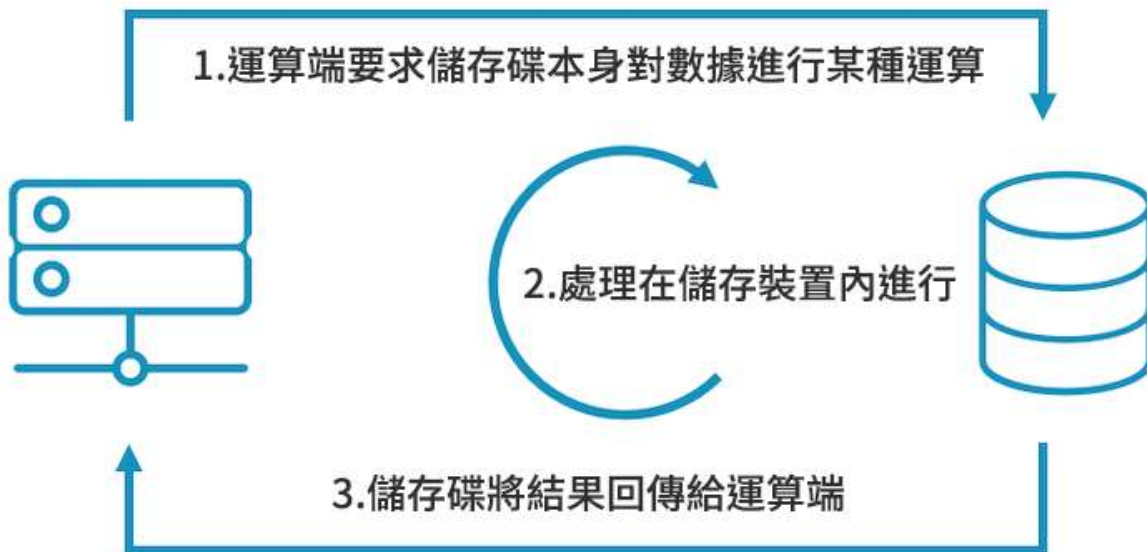
運算型儲存如何運作？

CSD 是一個提供持續數據儲存與運算服務的儲存裝置。運算型儲存與結合運算與儲存有關，在數據所在的本地運行應用，降低遠端伺服器需要的處理，並且減少數據的移動。為了做到這點，儲存碟的處理器專門直接在該儲存碟裡處理數據，讓遠端的主處理器能夠處理其它的任務。

傳統的儲存系統中，運算端想對數據進行某些處理：



運算型儲存系統中，運算端不會要求數據：



更多洞察，請閱讀我們的 [運算型儲存指南](#)。

Linux 促成運算型儲存

落實運算型儲存有多種方式，不過，主要的要求是把處理能力嵌入儲存碟控制器，以便運行像 **Linux** 與軟體元件等受到信任的作業系統。這點有下列幾個關鍵好處：

開放原始碼軟體，具有廣大的 **Linux** 開發者社群

具有廣大 **Linux** 開發者社群的開放原始碼軟體以及業界廣泛使用的標準工具，讓整個開發經驗更加簡便。能夠打造作業負載，並讓開發人員使用 **Linux** 架構的標準系統把它部署到儲存碟，同時遵循 **SNIA** 標準，可以讓系統簡化，同時也有利於更簡便的軟體開發。

立即可用的工具

有了 **Linux**，我們可以取用龐大的工具與開放原始碼軟體的生態系，以便於開發、部署與管理運算型儲存作業負載。這可以讓開發人員社群快速地把任務轉移到運算型儲存碟。

促成智能的儲存

在標準的 **NVMe** 碟中，儲存碟收到數據區塊，然後把它分解，並儲存在 **NAND** 晶粒的記憶體頁內。伺服器會要求收到一個數據區塊，然後從 **NAND** 取得它，並把它重組成一個資料段，最後再傳給主機。不過，儲存碟仍然不知道這些資料段會構成一個 **JPEG** 影像，比方說因為它並不瞭解該檔案系統。另一方面，在儲存碟運行 **Linux** 可以促成聰明的儲存，原因是它可以裝上標準的檔案系統，同時 **CSD** 應用可以瞭解數據塊確切代表何種檔案，並直接對數據執行動作。

將儲存碟視為迷你伺服器

在儲存碟運行的 **Linux** 可以管理這個儲存碟、發展作業負載，並下載新的作業負載，而且這一切都是使用現行的標準開放原始碼系統。它讓儲存碟以最低可能的成本，搖身變成一個迷你伺服器。

現在您可能會想：**Linux** 真的適合運算型儲存嗎？答案是肯定的。

它會不會太大？答案是：不會。

今日的儲存裝置已經擁有好幾個 **GB** 的 **RAM**，以及數 **TB** 的儲存容量與快速的運算力，可以處理大量的數據從儲存碟移進與移出。**Linux** 可能讓人想到針對大型伺服器做的大量軟體安裝，而且並未針對終端裝置上的儲存與運算進行調適，但 **Linux** 的需求與大型伺服器相比小了很多。軟體的大小也能顯著縮小。

有了 **Linux**，我們不再需要顯示器驅動程式，有好幾個功能也不再適用，而且您可以把它簡化，並針對控制器進行客製。例如，**Debian 9** 只需 **512MB** 的 **RAM** 與 **2GB** 的儲存空間。

我們可以使用用在這些複雜系統的標準開放原始碼工具，來實施 **CSD** 的管理。我們可以使用如 **Kubernetes**、**Docker** 或延伸的 **Berkeley Packet Filter (eBPF)** 等常用工具，安全下載與管理作業負載，達成應用或指令碼的安全執行。

Arm 提供最簡單、最快速、最符合成本效益的運算型儲存解決方案

Arm 的儲存解決方案提供簡單、快速且符合成本效益的技術、支援與龐大生態系，讓您成功部署運算型儲存。

Arm 處理器的產品選項

Arm Cortex-A 處理器，針對儲存裝置複雜運算任務的低耗電與高效能進行優化。此外，Cortex-R82 處理器則針對高效能即時與高階作業系統應用優化。這些具應用執行能力的處理器，可以利用下列促成運算型儲存：

- 可用來運行 Linux 的記憶體管理單元(MMU)
 - 為 ML 作業負載提供 Neon 支援
-

應用的移植與優化

Arm 與合作夥伴已經移植並優化所有主要的 Linux 配銷版本，以及開放原始碼應用。這部份會自動進行維護，並可以使用任何的 Cortex-A 處理器或 Cortex-R82 運行，而且不需要調適。Arm 同時也在內部與透過 Linaro 優化這些應用，以確定所有的東西都能最佳化地在 Arm 上運行。

工具與程式庫的軟體生態系

有了 Arm 軟體生態系的支援，程式設計的工作量可以降到最低。在 Cortex-A 處理器與 Cortex-R82 上運行的 ML 軟體程式庫，可以加速影像或其它檔案的搜尋速度。

Arm 的合作夥伴 NGD Systems 公司，正在研究使用運算型儲存來協助航空公司提升他們的飛航數據分析。航空公司今日每個小時都會產生數個 TB 的遙測數據，而卸載與分析這

些數據需要花上好幾個小時，但他們卻沒有這麼多的時間。有了運算型儲存，飛航分析方法可以在正確的時間提供給正確的人，協助提升飛航安全。

CSD 也可能運行許多其它非 Linux 的運算類型。針對例如加密、壓縮或重複數據刪除等固定目的運算功能，我們都可以在 CSD 系統上使用低階的即時軟體、硬體加速或神經處理器(NPU)。這些特定功能與加速器類型，可以內建在 CSD 內，並透過業界定義的 CSD 協定延伸直接存取。

這些低階功能，也可以透過可取用的高階作業系統進行存取。高階作業系統提供的彈性與客製化的簡便性，結合龐大的開發人員社群與低階加速器，可以達成非常高效能與高效率的 CSD 解決方案。

運算型儲存的未來已經降臨

目前許多合作夥伴都已供應以 Arm 處理器架構設計的裝置，同時業界也在努力要讓所有儲存裝置的開發人員與參與者實現保持一致。Arm 主動參與 [SNIA 儲存網路產業協會](#)，與 45 家公司及 202 個會員共同致力於定義標準。這個標準由於可以消除碎片化的風險與相容性的缺乏，將有助運算型儲存的採用與開發。

請造訪我們的[運算型儲存網站](#)，以瞭解更多有關 Arm 的儲存解決方案。