

## 1 History of kernel

自從 2014 年以來，container 席捲了全球，unikernel 是許多人的下一個 container，2016 年 1 月，主力在 VM 的 Docker 併購了 Unikernel System，更讓人意識到「無核化」的 VM 將可能是未來的趨勢。

Unikernel System 的代表作是此篇論文提到的 MirageOS。內核，代表的是掌管作業系統的一個概念。若要順利執行應用程式，必需和硬體合作，讓應用程式直接操作硬體不會是我們樂見的，而內核即是在應用程式和硬體間提供了一層抽象概念。例如把硬碟抽象成了文件，這樣 user 就能透過 file system 進行管理。傳統的內核會將所有的硬體抽象化實現在其中，代表作就是 Linux，這樣的內核被稱為 Monolithic Kernel，最大的缺點就是管理不易，並且其中一部分產生的 bug 會使整個內核崩潰。

Micro Kernel 的出現一定程度的解決這樣的問題，在內核中只保留必要的 module，例如：記憶體管理、CPU 調度、IPC 等等，其它如：file system、Network I/O 都移到 user 端來處理，這樣的好處是內核變小了，所以較不容易崩潰，但因為各個 module 需要以 IPC 的方式溝通，因為會有一定 overhead，效能不如 Monolithic Kernel，經典的例子有 Mach、Mac OS X。後來又有了 Hybrid Kernel，將不常用的內核 module，或是本來 IPC overhead 就不是太誇張的功能，移出內核。

## 2 Contribution

”A unikernel is a specialised, single address space machine image constructed by using library operating systems.” 是維基百科給出的一段解釋，unikernel 最大的貢獻就是沒有 user space 和 kernel space 分別，只有一個連續的地址，因此 unikernel 一次只能執行一個應用程式，所有操作軟體、硬體的邏輯，都在一個空間之中。這樣乍看之下像是個閹割版的內核，但 unikernel 在高度專門化的設計下，能達到非常小的 image，就是因為夠小，deployment 是 unikernel 的強項，unikernel 的產物是一個 kernel image，可以在 Hypervisor、Bare Metal 等等環境執行。

Mirage 一個 Standard build 的 DNS 才 0.449 MB，Web Server 也才 0.673 MB。Unikernel 編譯過程時，允許應用程式開發人員使用動態參數來配置各個 library，這樣大大減少了配置複雜的多服務應用程式 VM 所需的工作量。

輕量的 unikernel 也在 booting 達到很快的水準，比傳統內核快多了，而輕量化的程式碼使得 unikernel 中沒有 shell 可用，也沒有密碼文件，這樣也相對安全許多。

## 3 Conclusion

MirageOS 的 unikernel 在 memory size、# threads、throughput 等都贏了 Linux，而且模組化的程式碼，輕量的同時，也更好維護，這些都是 unikernel 具有前瞻性的原因。

我們可以發現，unikernel 在軟體開發過程與傳統最大差異在於測試。傳統的方式可能發布一個應用程式、一個 container image，而 unikernel 則是一個高度客制化的 kernel。將 kernel「專門化」有其優點，但也有他的缺點，就像 Micro Kernel 那樣，有可能會在 IPC 造成過大的 overhead，另外由於 MirageOS 是使用 OCaml、C/C++ 和 ASM 混合著編譯的，所以也必需要求開發者對 OCaml 的熟稔度夠高。若我們在 Unikernel 裡執行一個 Docker Container，想法看似美好，但同樣也有許多問題需被解決，這樣沒有利用到容器容易分發的優點。而且 MirageOS 的 unikernel 實作方式有一個大缺點是他高度依賴了不安全的程式碼，如何基於這些不安全的程式碼，繼續往下開發，或是乾脆整個打掉重練，這些都是在未來會面對到的挑戰。