



Anil Madhavapeddy et al. :

Unikernels : Library Operating Systems for The Cloud

ACM ASPLOS (2013)

OS 研究の不思議

オペレーティングシステム (OS) やシステムソフトウェアの研究者は一体何の研究をしているのだろうか? と疑問を抱かれるのではないかと思う。OS は依然として脆弱性などの問題を抱えており、アプリケーションソフトウェアから見れば、OS の設計や実装は枯れていて、特に大きく進歩しているようにも思えない。しかしながら、OS の世界では、新しいアーキテクチャの検討が進行中である。

クラウド時代のライブラリ OS

ライブラリ OS は、モノリシックカーネルやマイクロカーネルに次ぐ、新しい OS 設計の模索である。その特徴は、OS をライブラリとしてアプリケーションに組み込み、単体でマシン起動可能とする点である (図-1)。

ライブラリ OS 自体はまったく新しい概念ではない。1990 年代の Exokernel や Nemesis など古くか

らあるアイデアである。今回紹介する Unikernel に代表される研究が画期的なのは、クラウドコンピューティングが広く普及したので、仮想化ハイパーバイザの存在を仮定する点である。クラウドの時代においては、仮想マシンの作成や破棄が容易となり、アプリケーションごとに仮想マシンを立ち上げることも簡単である。また、従来のライブラリ OS の大きな弱点であったさまざまなハードウェアのサポートも、仮想マシンの中ではエミュレートされた仮想デバイスとして標準化されているため、大きな問題とはならない。

本論文の Unikernel, そのプロトタイプ実装としての MirageOS は、まさにクラウド時代のライブラリ OS である。また、単一アドレス空間上でライブラリ OS とアプリケーションをリンクし、マシン起動可能としたものに Unikernel と名付け、専門用語として広めたのも著者らの貢献であると言える。

クラウド世代のライブラリ OS, Unikernel に共通する特徴は次のとおりである。

- 単一目的のアプリケーション : 仮想マシン 1 台

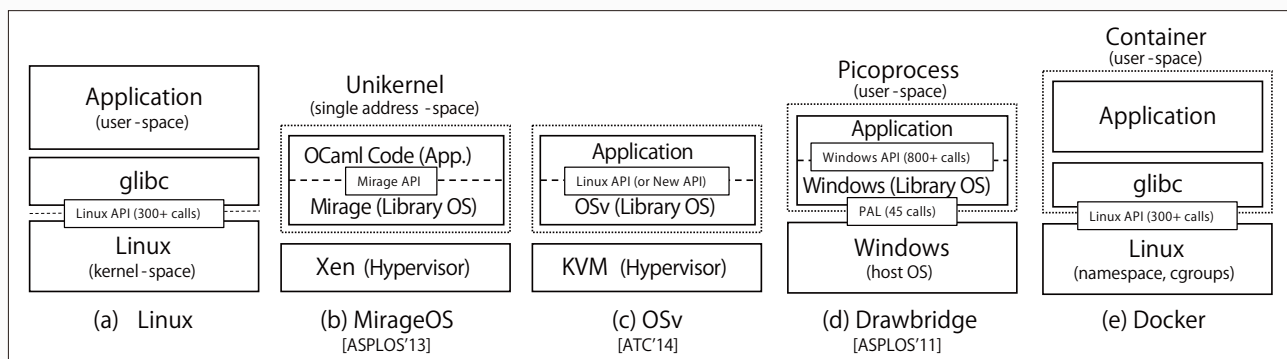


図-1 OS 設計の比較



あたり起動するのは、1つのアプリケーションのみとする。近年のクラウドシステムは、Webサーバ、データベースなどの複数のアプリケーションで構成されている。そのそれぞれに仮想マシンを割り当てる。従来のOSでは、マルチユーザ、マルチアプリケーションを想定していたのに対して、単一ユーザ、単一アプリケーションを仮定する。

- 単一アドレス空間：従来のOSでは、ユーザ空間とカーネル空間を明確に分離し、実行していた。ライブラリOSでは、単一アドレス空間上でアプリケーションとカーネルを共存させる。従来OSのシステムコール呼び出しは単なる関数呼び出しとなる。システムコール呼び出しに伴うオーバーヘッドは大幅に削減され、性能が向上する。
- TCB (Trusted Computing Base) の削減：近年のモダンOSは非常に巨大である。OS分野のセキュリティでは一般にコード量を削減するアプローチが有効である。ライブラリOSでは、冗長な抽象化レイヤを削減し、全体をコンパクトにすることができる。これはセキュリティの向上だけではなく、起動時間の短縮、メモリやディスク使用量の削減にも繋がる。

一方で Unikernel の実現には、いくつかの設計上の選択肢があり、MirageOS では下記の選択を行っている。

- 高水準言語の利用：新しいOSを開発する上で大きな課題となるのが、既存のアプリケーションとの互換性である。MirageOSでは、この従来のアプリケーションとの互換性をきっぱりと諦め、OCamlで記述されたアプリケーションのみを対象とする。強い静的型付けを備えた高水準言語の利用は、過去のSingularityの研究とも関連している。また、本論文が採択されたASPLOSは、OSとアーキテクチャ、プログラミング言語分野の境界領域のトップ会議であり、その点でも本論文の採択

は適切である。ここで面白いのは、ほぼ同時期に登場したOSvでは、LinuxのABI (Application Binary Interface) との互換性を重視していたのに対して、MirageOSでは言語的なメリットを重視しており、まったく正反対のアプローチとなっている。

- コンパイル時特化 (specialization) : OCamlのような静的特化をサポートする言語を用いれば、OSとアプリケーションを結合し、大域的な最適化や不要コードの削減を実施することが可能である。また、コンパイル時に設定情報も静的に埋め込めば、変更や改竄は不可能となる。これは一時期流行したイミュータブルインフラストラクチャの概念とも関連している。

一般にOS分野の評価では、マイクロベンチマークとマクロベンチマークの2つを実施するというのがお約束である。マクロベンチマークでは、DNSサーバ、OpenFlowコントローラ、WebとDBの統合サーバを実装し、評価しているが、既存の本格的なアプリケーションと比較すると機能は最小限であり、かろうじて動いていたというのが実情だろう。

その後の展開について

MirageOSはライブラリOSの中でも極端なアプローチをとったため、広く普及するには至らなかった。一方で、論文としては非常に面白く、ライブラリOSの可能性と限界を示しているといえる。実際、MirageOSの開発元スタートアップであるUnikernel Systemsは2016年にDocker社に買収されたが、現在までその成果が広く知られるに至っていない。

これまでのライブラリOSはあくまで研究用OSのポジションであり続けた。Drawbridgeで実現されたピコプロセスを除き、ライブラリOSはほとんど実用化には至っていない。一方で近年、既存のLinuxのABIと互換性を保ったライブラリOSが複数登場してきている。つまりは当初のOSvに近

い研究が増加している。経験の蓄積も大幅に進んでおり、ライブラリ OS は実用化目前かもしれない。ピコプロセスは、既存のホスト OS 上に最小限のマシン抽象化インタフェースのみを設け、Windows や Linux などライブラリ OS として実行する方式である。これは、ホスト型ハイパーバイザに近いアプローチであり、ベアメタル型に近い MirageOS や OSv とは異なる。

今回の執筆依頼にあたり、ACM SIGOPS の The Hall of Fame Award など調査したが、OS のコミュニティでは評価を得るまで最低 10 年間、実際には 20 ~ 30 年以上の時間を要するようである。ライブラリ OS の登場は 2010 年代前半であるので、そろそろ本格的に実用化されてもおかしくはない。また、IoT やエッジコンピューティング、マイクロ

サービスなどのライブラリ OS 向きの新しい文脈も登場している。ライブラリ OS は単一アドレス空間でコンパクトであることから、学生や若手研究者にも理解しやすく、参入しやすいであろう。私の本来の専門は OS よりやや上層のシステムソフトウェアであり、私自身がナナメ読みをしてしまっているかもしれないが、今後の OS 研究の発展に期待していただきたい。

(2020 年 7 月 31 日受付)

.....

杉木章義 (正会員) sugiki@iic.hokudai.ac.jp

2007 年電気通信大学電気通信学研究科博士後期課程修了。博士(工学)。科学技術振興機構 CREST 研究員、筑波大学学術情報メディアセンターを経て、北海道大学情報基盤センターに着任。システムソフトウェアと OS に関する研究に従事。

