Unikernels

Boris TEABE
boris.teabe@inp-toulouse.fr

Goals

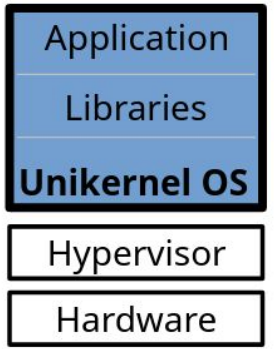
1. Bare Metals
2. Virtualization
3. Containers
4. **Unikernels/MicroVM**

Unikernel

- **Unikernel: application + dependencies + thin OS compiled as a static binary running on top of a hypervisor**
- **Single-***
  - Single purpose: run 1 application
    - Want to run multiple applications? run multiple unikernels
  - Single process
    - Want to run a multi-process application? run multiple unikernels
    - However, SMP (multicores) and multithreading are supported
  - Single binary and single address space for application + kernel
    - No user/kernel protection needed

Unikernel

- **A form of lightweight virtualization**
  - Contain and run only what is absolutely necessary to the application
  - Cost advantage: memory/disk footprint reduction
  - Considered as a secure alternative to containers
    - Unikernels are virtual machines!
- Per-application tailored kernel (LibOS/Exokernel model)
  - Specialization for lightweightness but also performance
- Reduced OS noise, increased performance
  - Low system call latency: app + kernel in ring 0, system calls are function calls
  - Sub-second boot time



Unikernel

- Cloud applications: servers, micro-services, SaaS
- Embedded virtualization, Edge computing, IoT
- Network Function Virtualization, HPC, VM introspection, malware analysis, secure Desktop applications
- etc.
- Contrary to containers which are a mature and widespread technology, unikernels are still at the stage of research prototypes

Unikernel

- **Unikernels are lightweight AND secure, why didn't I heard of them?**
- Because it's hard to port existing applications!
  a. Proprietary software → source code not available
  b. Incompatible language
  c. Unsupported features
  d. Porting is hard, needs knowledge about both application and unikernel
  e. Complex built toolchains