Will Serverless End the Dominance of Linux in the Cloud?

Ricardo Koller and Dan Williams, IBM Research

Wosc2 2017

IBM

The beginning of the end of Linux

- “[T]hroughout the history of computer science there has been a fairly constant opinion that current operating systems are inadequate”
  - Engler and Kaashoek 1995

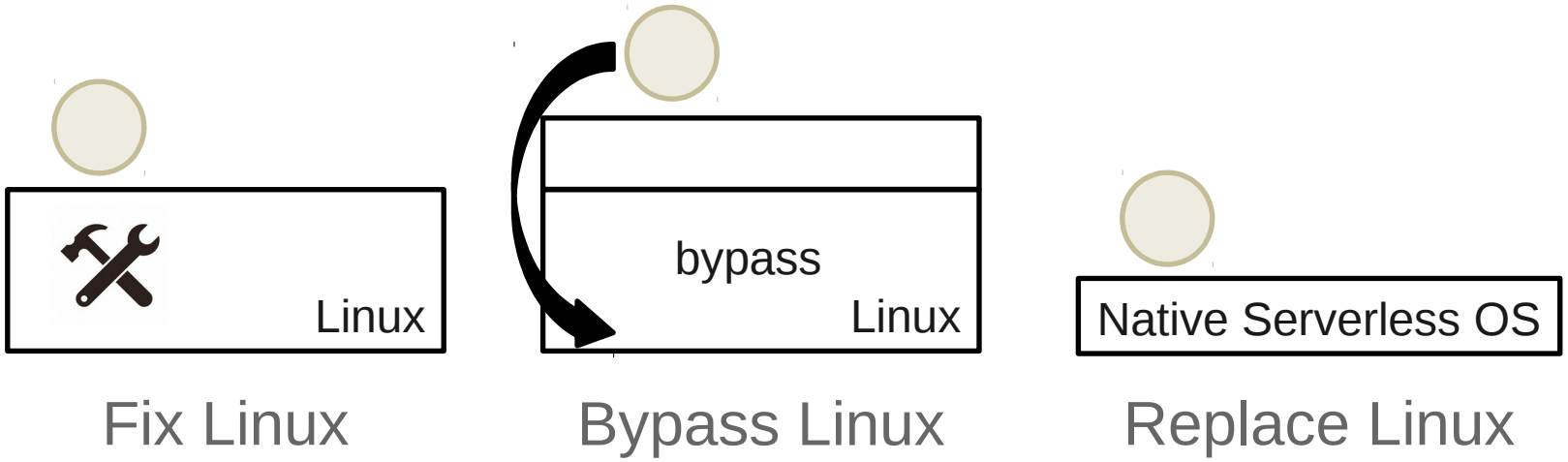
- Why now?

The beginning of the end of Linux

- “[T]hroughout the history of computer science there has been a fairly constant opinion that current operating systems are inadequate”
  - Engler and Kaashoek 1995

- Why now?
  - Cloud unit of execution shrinking: **serverless**
  - **Complexity** of kernel continues to grow

Options and Roadmap



- ~~Overview~~
- Serverless and its demands
- How Linux containers fail to meet these demands
- Rethinking the kernel

Serverless and its demands

- Isolation for multi-tenancy (excludes native processes)
- Performance

Serverless and its demands

- ~~Isolation for multi-tenancy (excludes native processes)~~
- Performance

Serverless and its demands

- ~~Isolation for multi-tenancy (excludes native processes)~~
- Performance

Latency

User: should launch immediately
Provider: should not require caching complexities

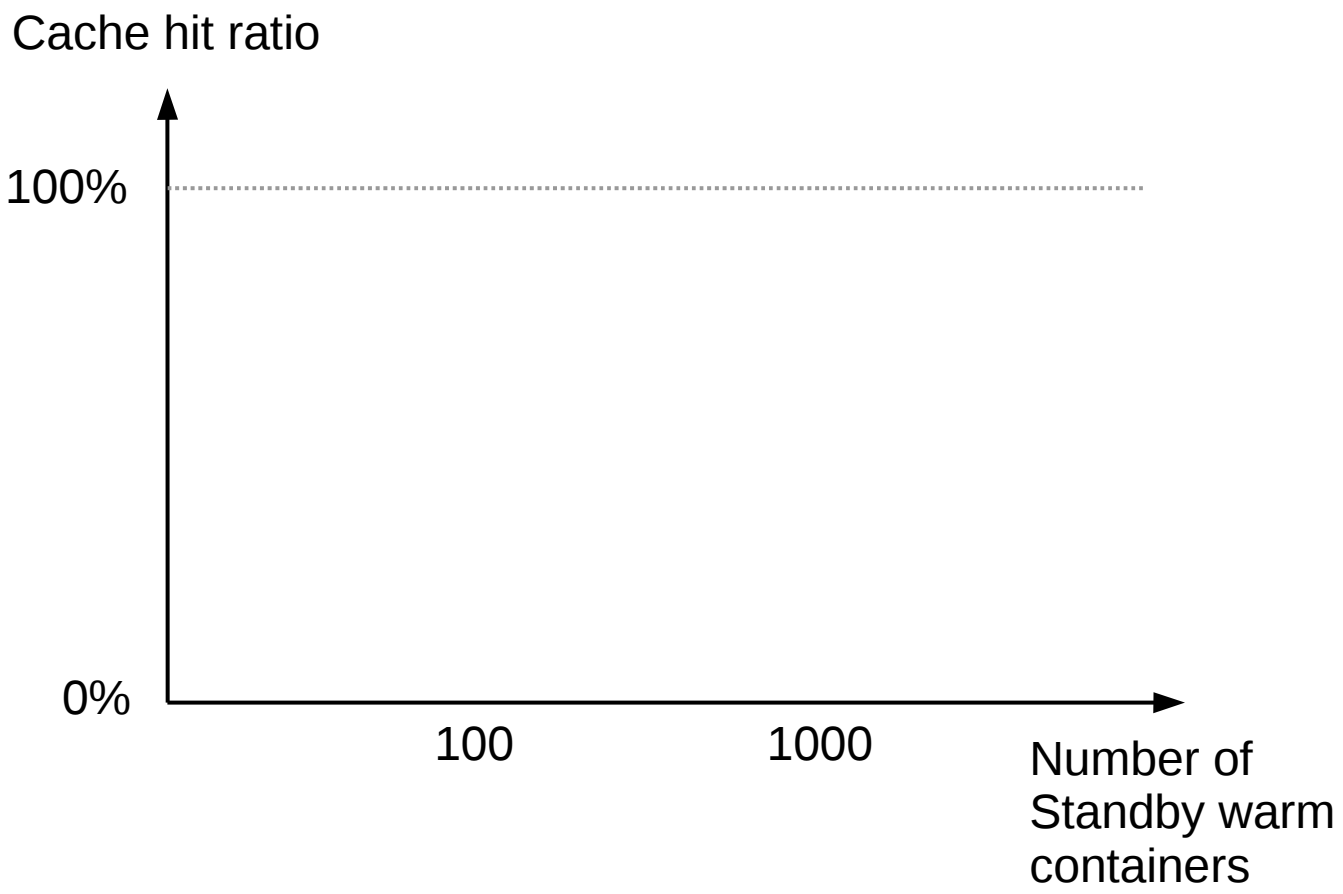
Target: **100 ms**

Throughput

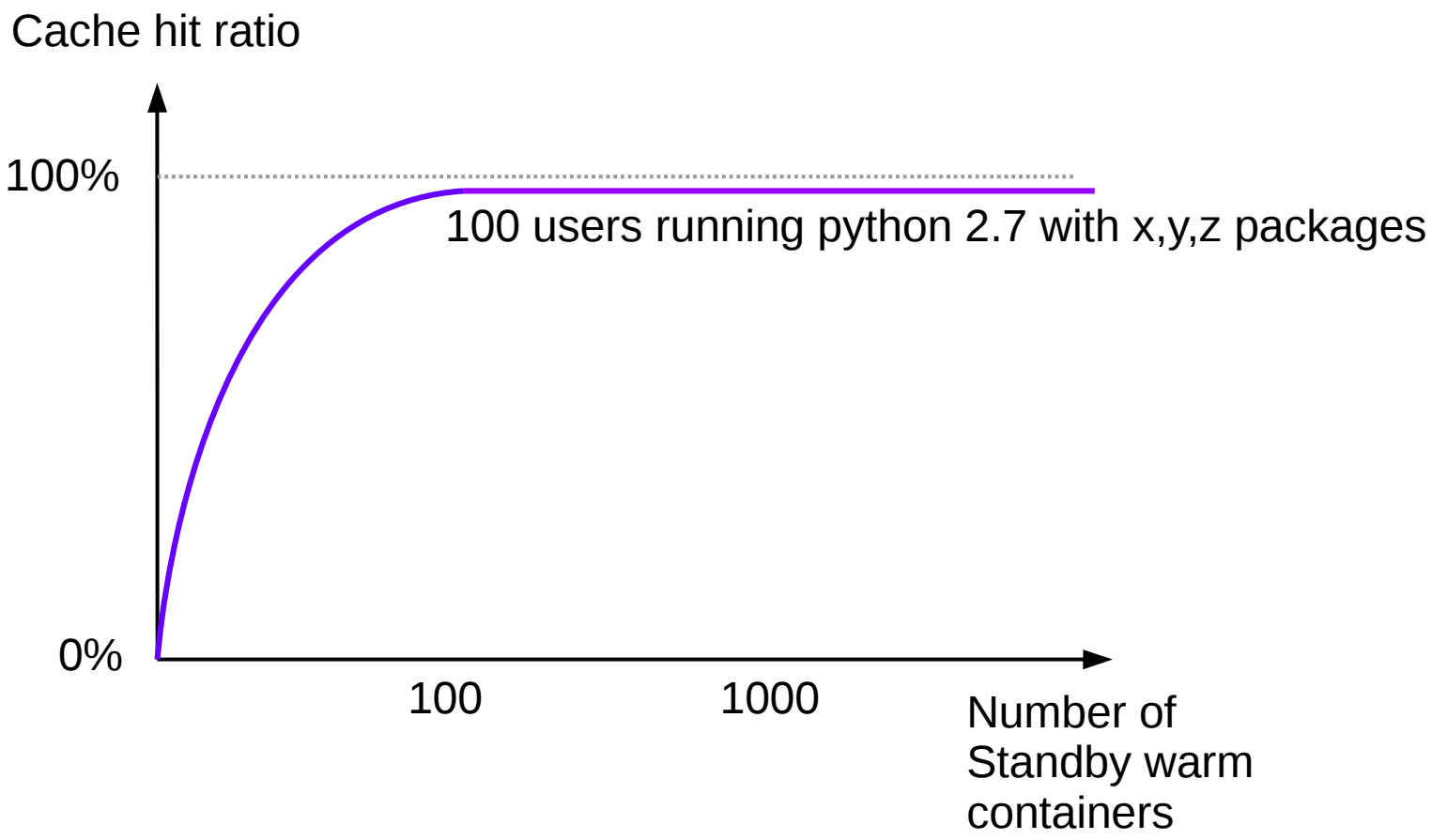
Provider: should at least cover hourly cost of server

Target: **125 actions/sec**

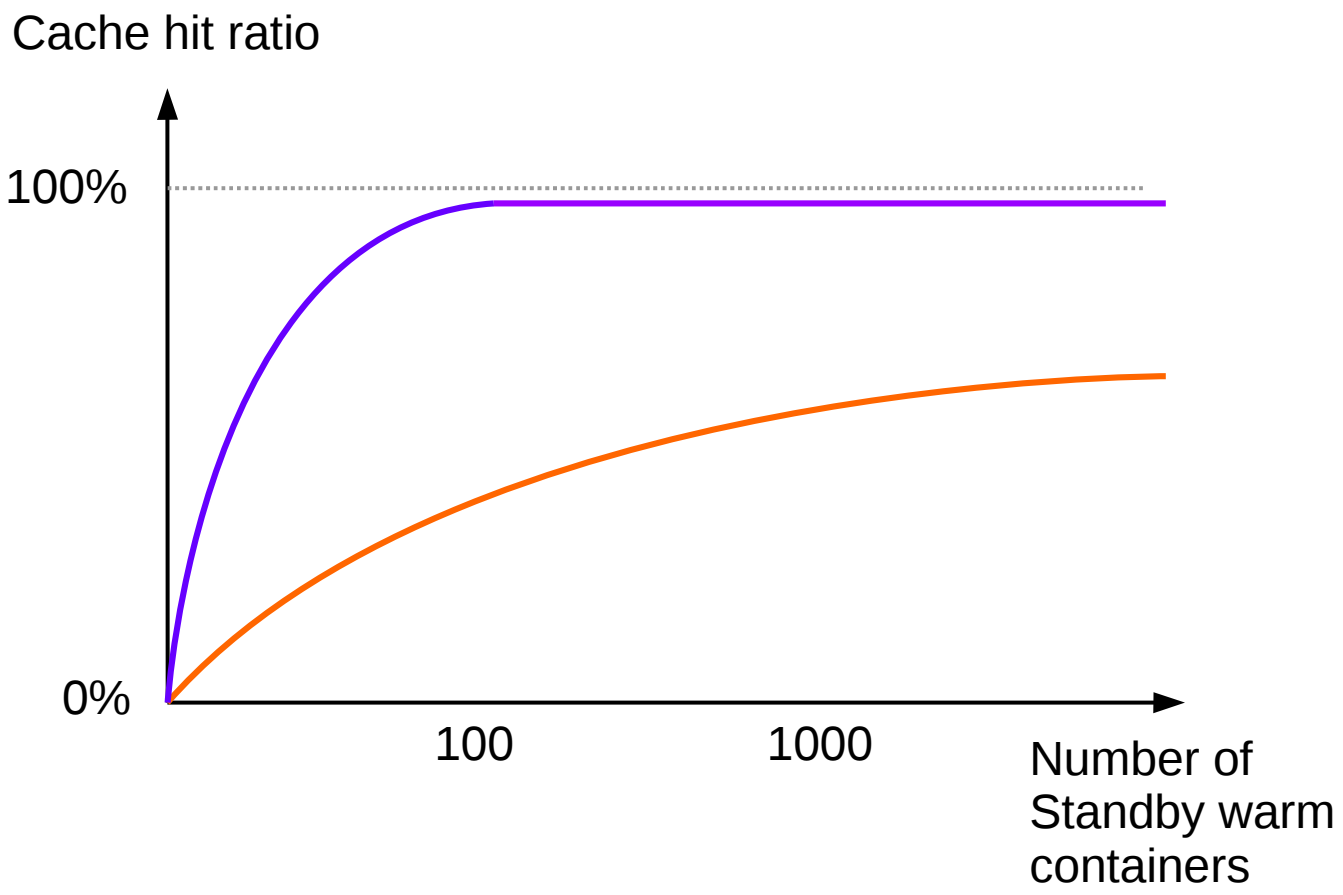
What about caching of warmed containers?



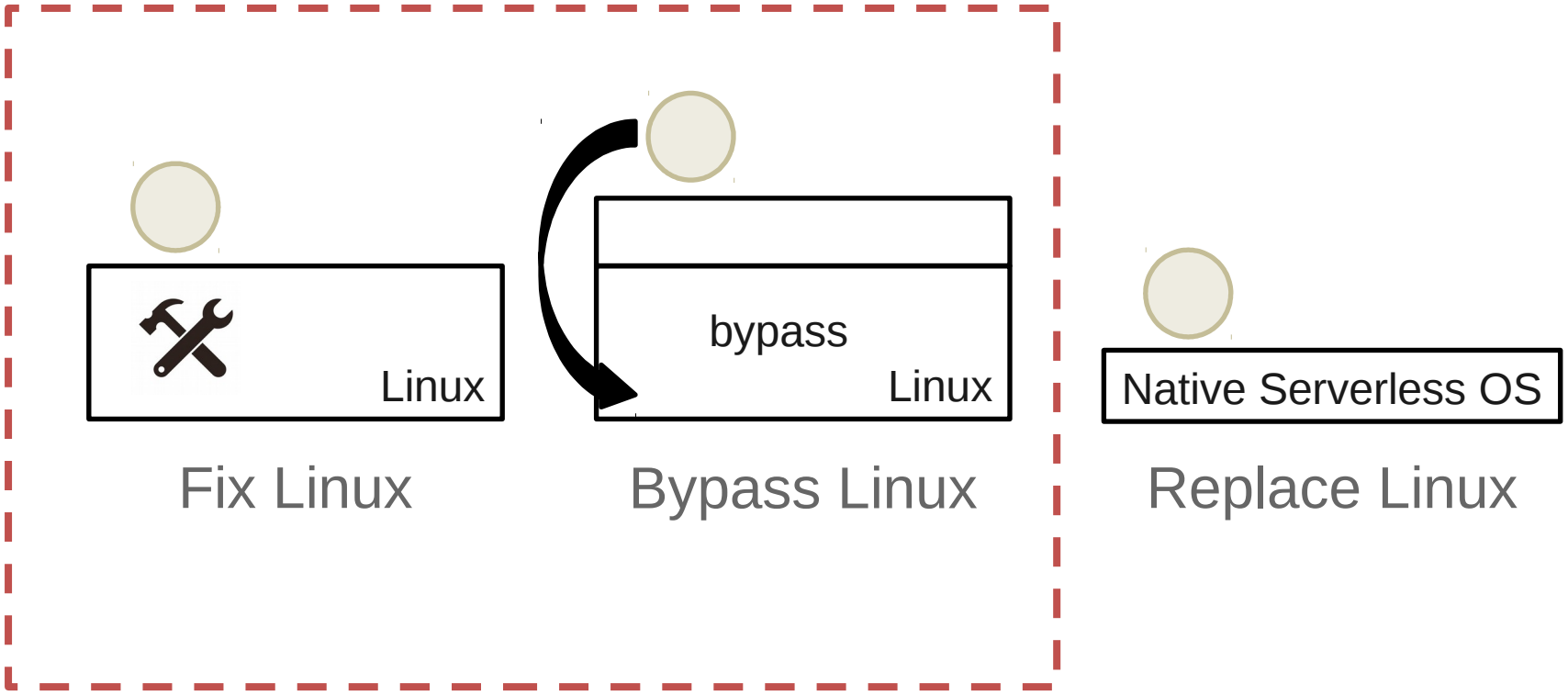
What about caching of warmed containers?



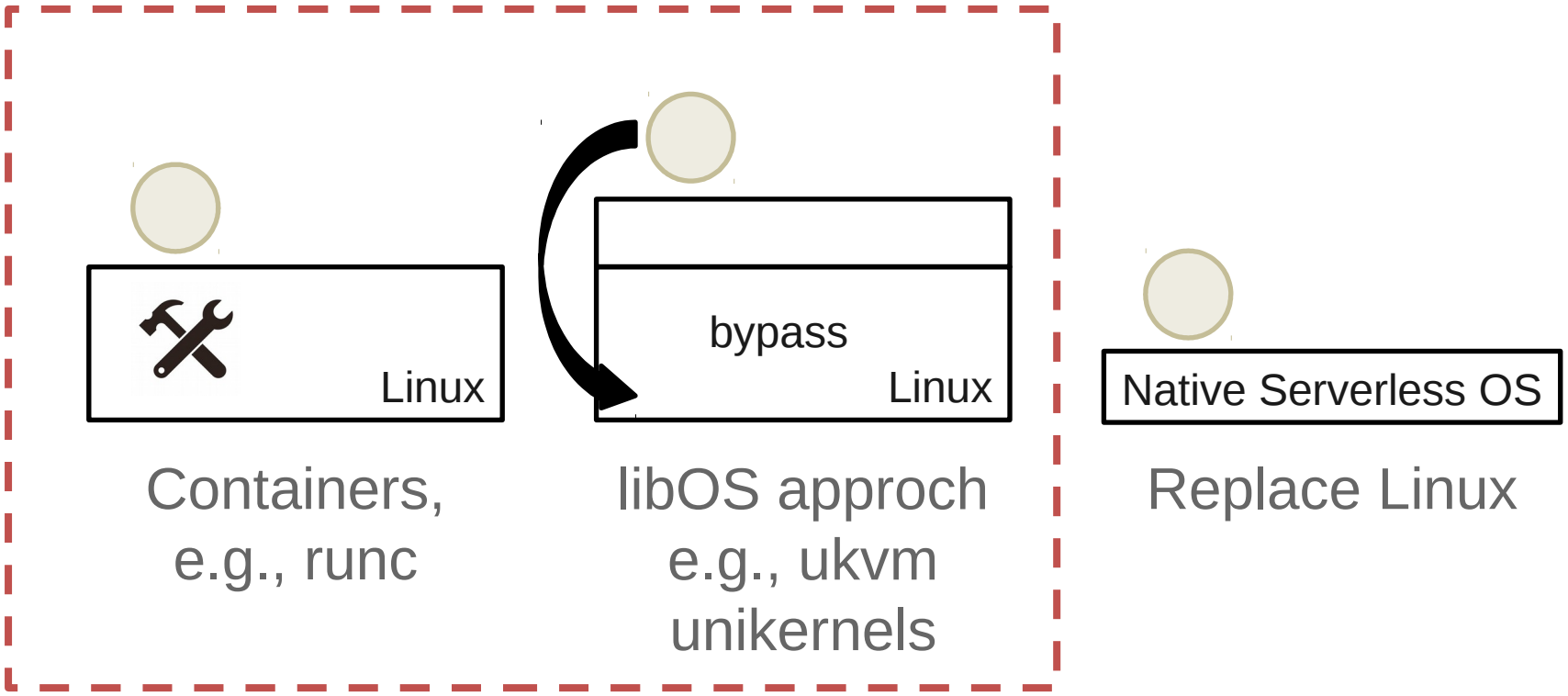
What about caching of warmed containers?



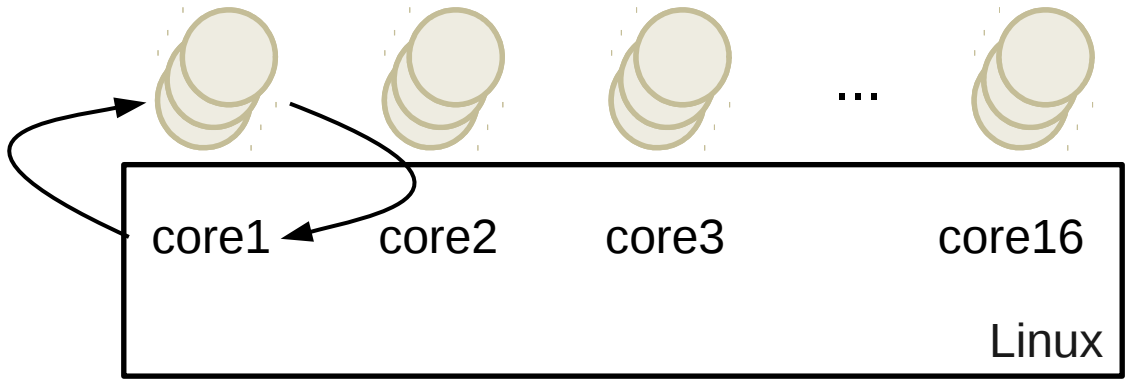
Options Revisited



Options Revisited



Experimental setup

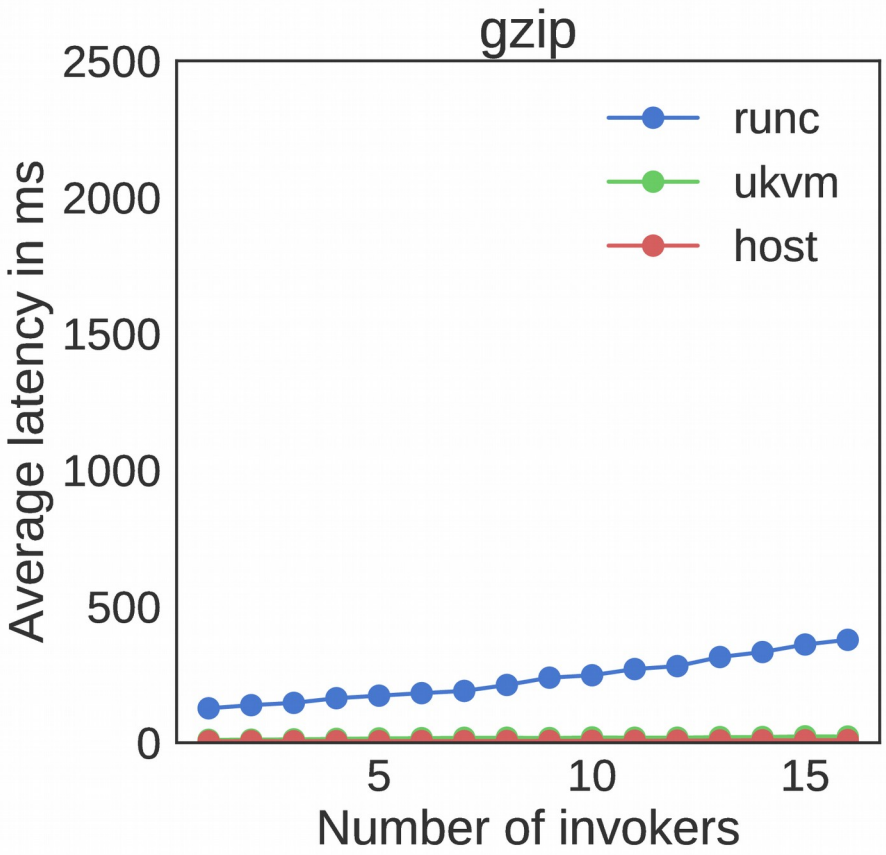


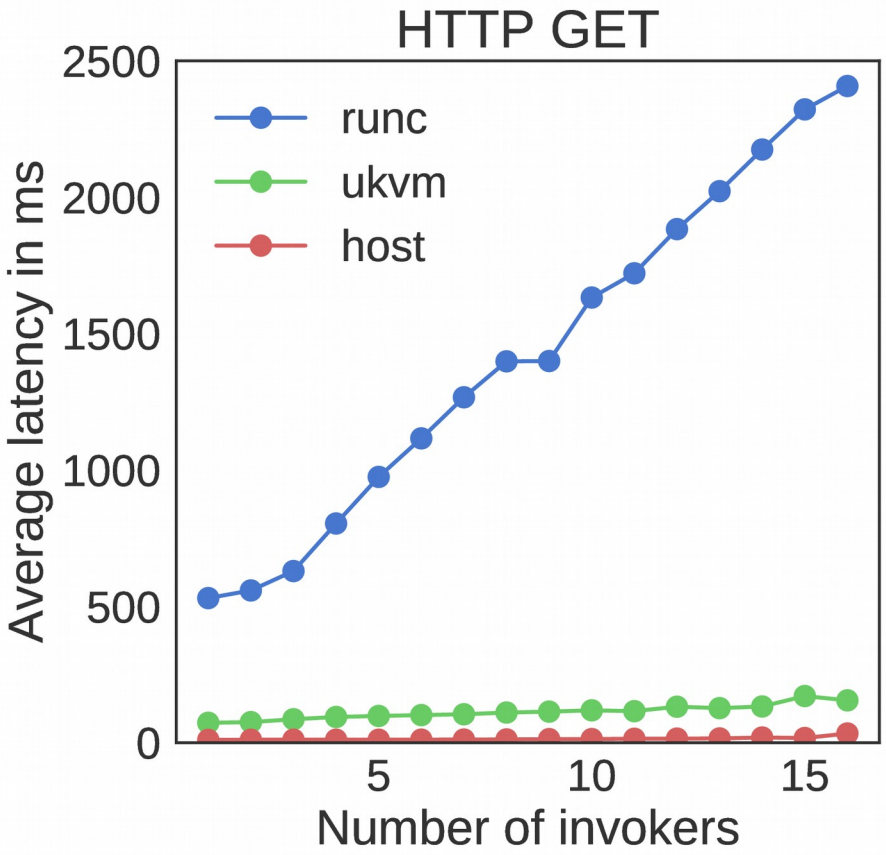
runc container

ukvm unikernel

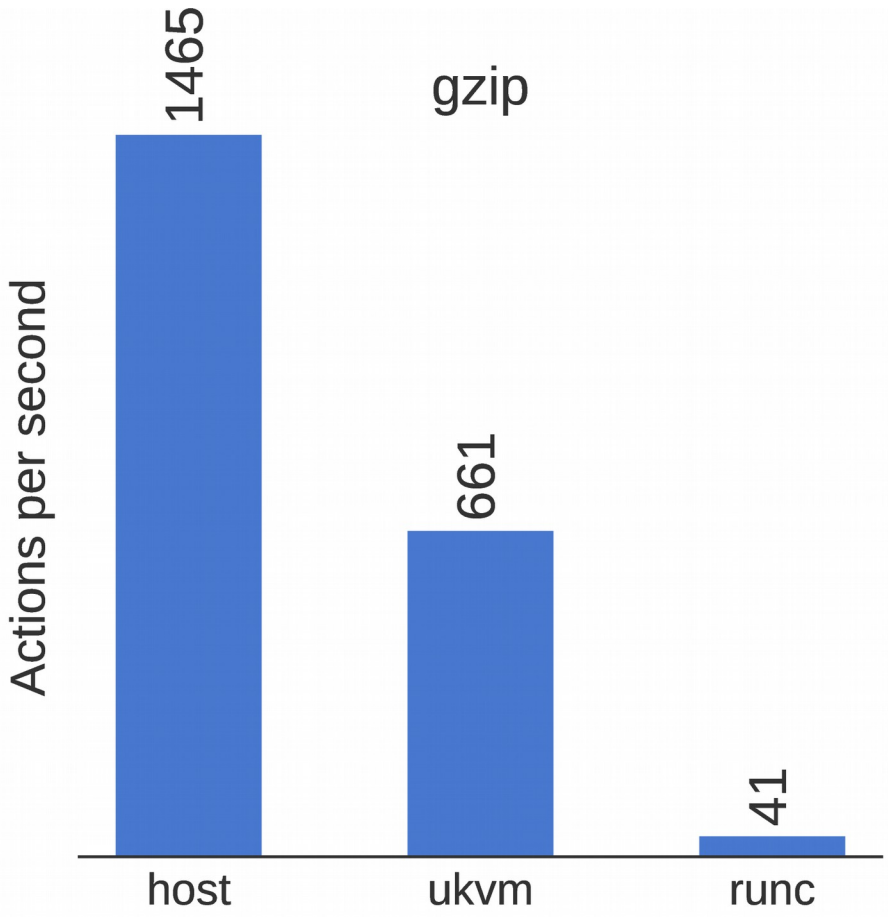
regular host process

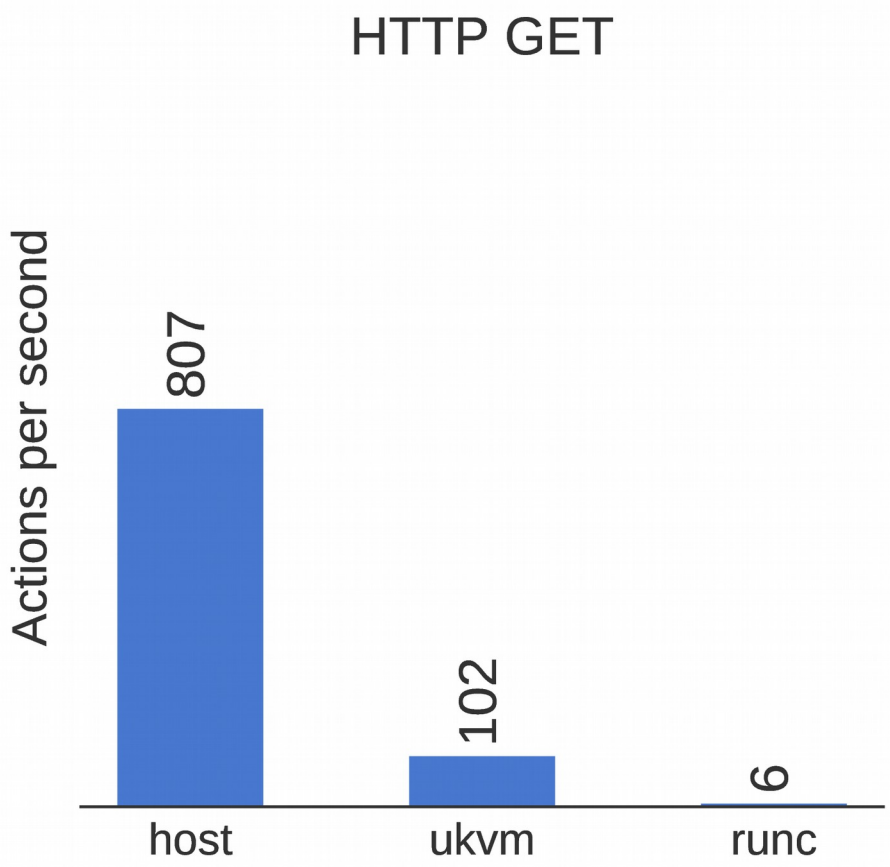
Latency



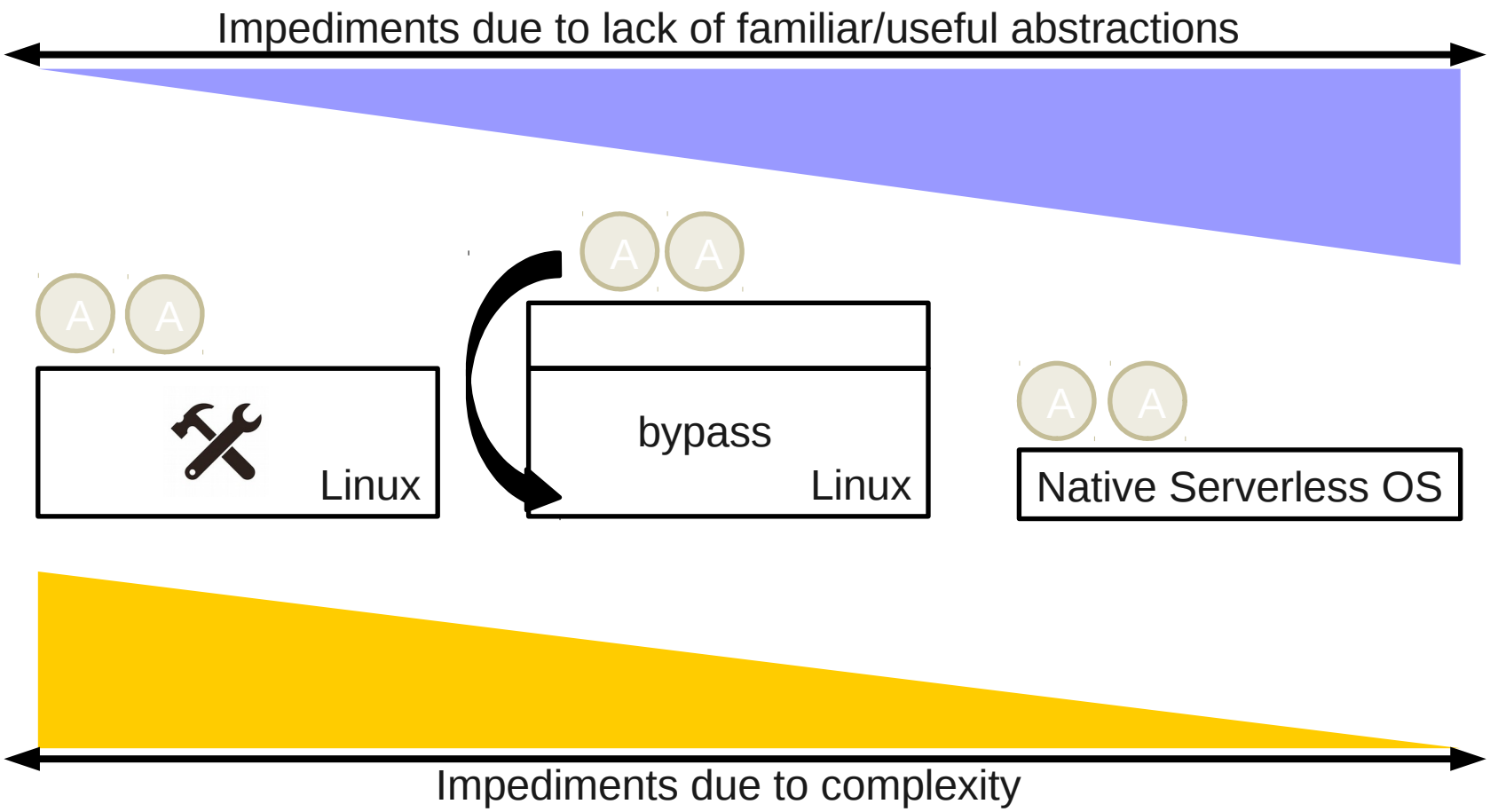


Throughput

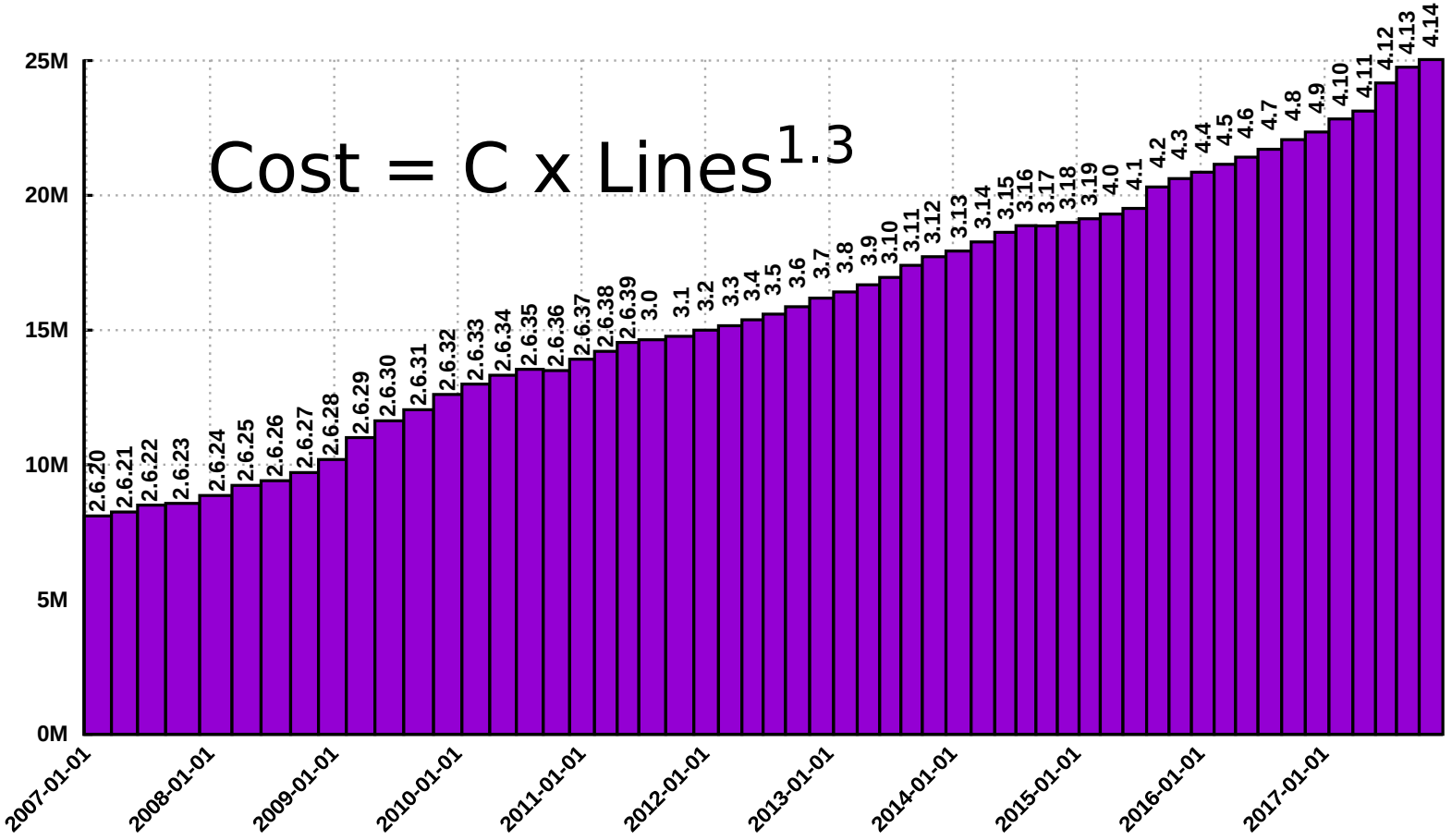




A tradeoff



Can we fix containers?

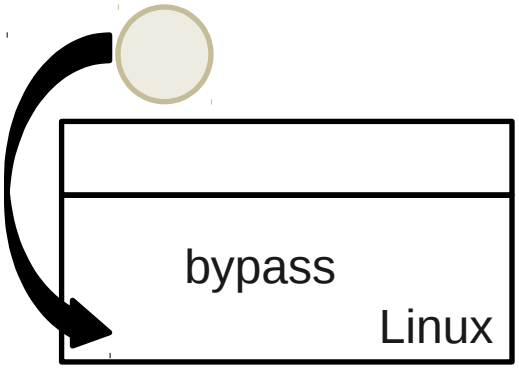


[1] https://upload.wikimedia.org/wikipedia/commons/1/11/Lines_of_Code_Linux_Kernel.svg

[2] The Linux kernel as a case study in software evolution. Journal of Systems and Software 83, 3 (2010)

If we choose to bypass Linux…

- Linux is just used for setup
  – “the OS is the control plane”
- We would prefer that the kernel **stop trying to adapt** to cloud workloads
- We should choose to **design** for bypass



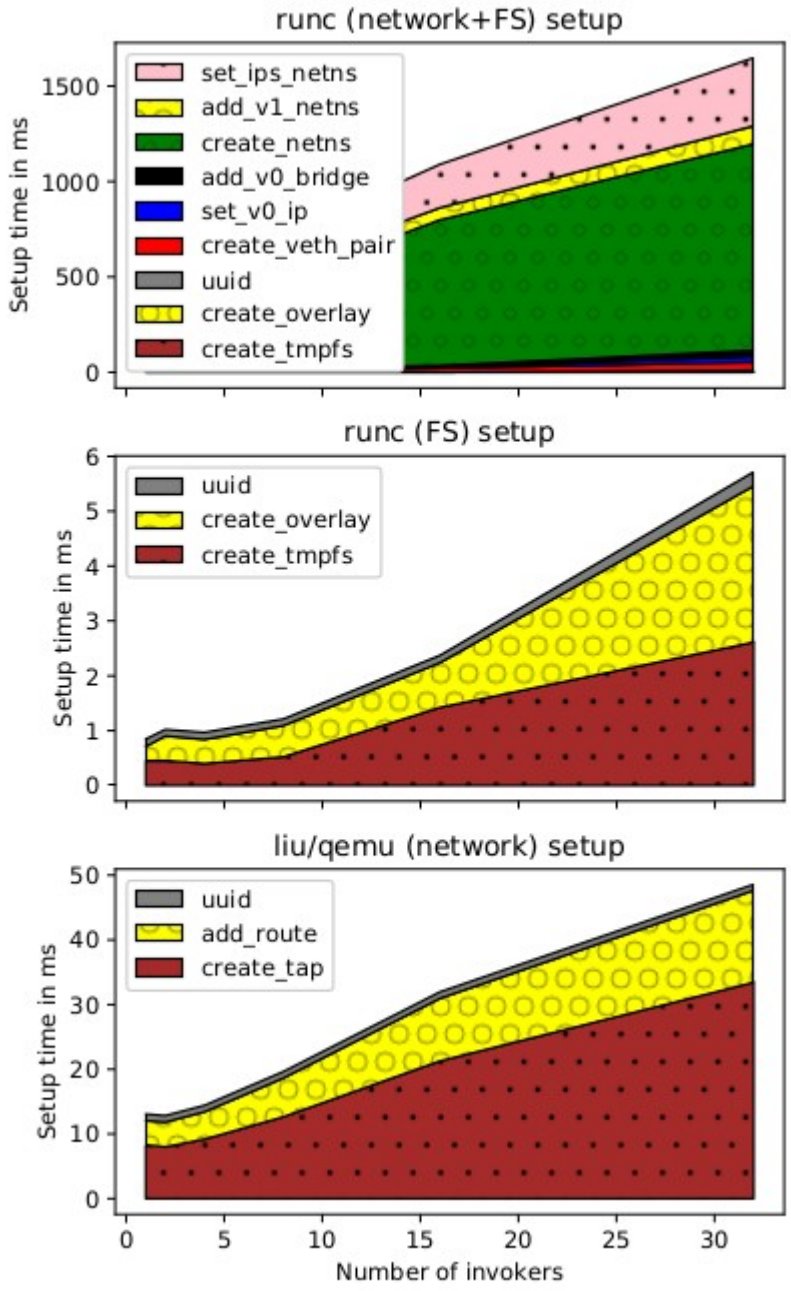
Bypass Linux

If we choose to replace Linux…

- No preemptible scheduling
  - actions are short and run-to-completion
- No process synchronization
- No IPC (inter-process communication)
- Limited set of I/O calls: literally input and output
  - files are unnecessary abstractions

Conclusions?

- Native abstractions in Linux (containers) are not suitable for serverless at this time
- Can they be fixed?
- Should we bypass Linux?
  - Then we should design for bypass
- Is it time to replace Linux? Can that even happen?



Cold starts