Will Serverless End the Dominance of Linux in the Cloud?

Ricardo Koller, Dan Williams
IBM T.J. Watson Research Center (HotOS ‘17)

Presented by Tai-Sheng Cheng

The Beginning of the end of Linux

- “Throughout 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
  - Complexity of kernel continues to grow

Why now?

- Complexity of kernel continues to grow
  - Major changes that introduce new abstractions to support containers
  - Next significant changes for Linux will take even longer [1,2]

- Cloud unit of execution shrinking
  - Current trend of cloud: lightweight applications rather than heavyweight systems
  - *Serverless* architecture: trend that lightweight applications will furthermore evolve into smaller lambdas or actions

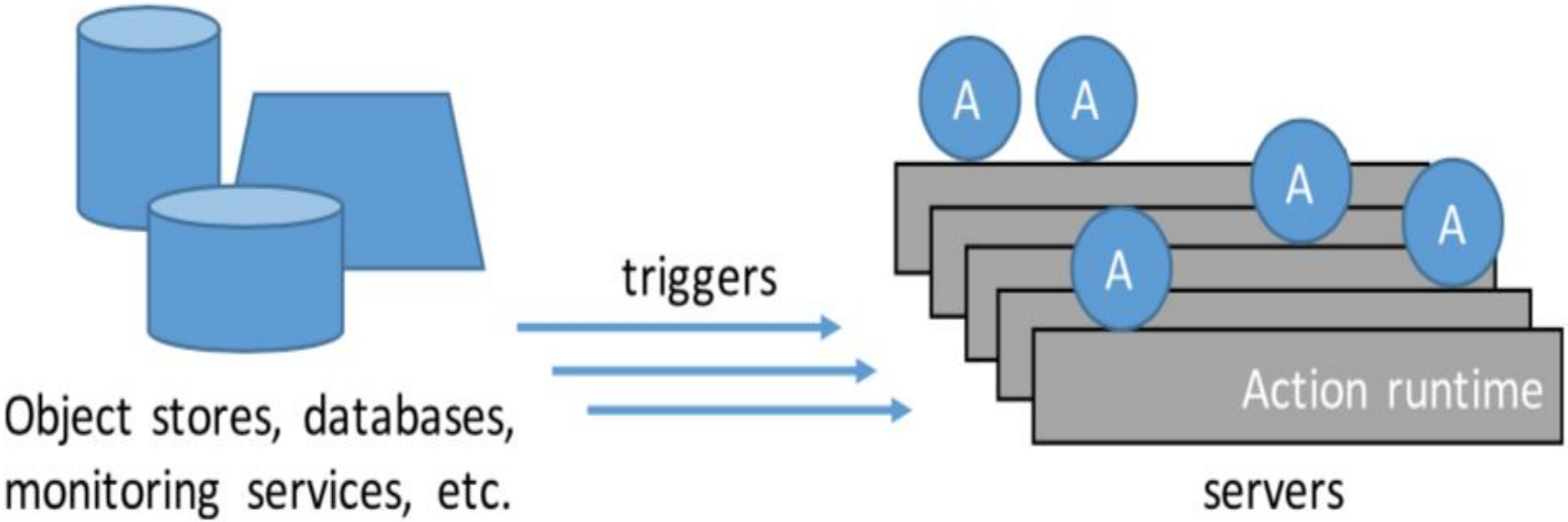
[1] Ayelet Israeli and Dror G Feitelson. 2010. The Linux kernel as a case study in software evolution.
[2] Dominik Strzałka. 2012. Fractal properties of Linux kernel maps. Computer Science and Engineering 2, 6 (2012), 112–117.

What is “serverless computing”?

- “Serverless” is not a good term
  - Doesn’t mean “no server”

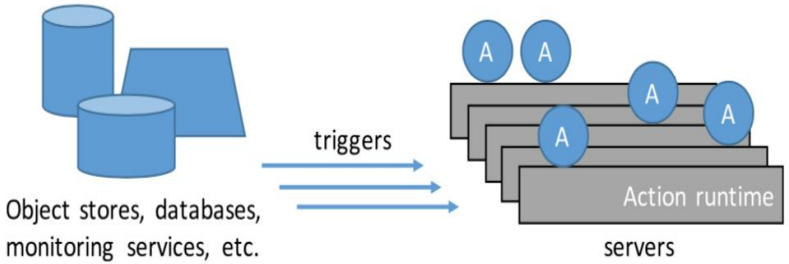
- “Serverless computing is a cloud-computing execution model in which the cloud provider acts as the server, dynamically managing the allocation of machine resources.”
  --- Wikipedia

Serverless Architecture



Serverless Architecture

- User-upload code (lambda, or actions)
- Granularity of actions are 100ms
- User only be charged when actions are executed

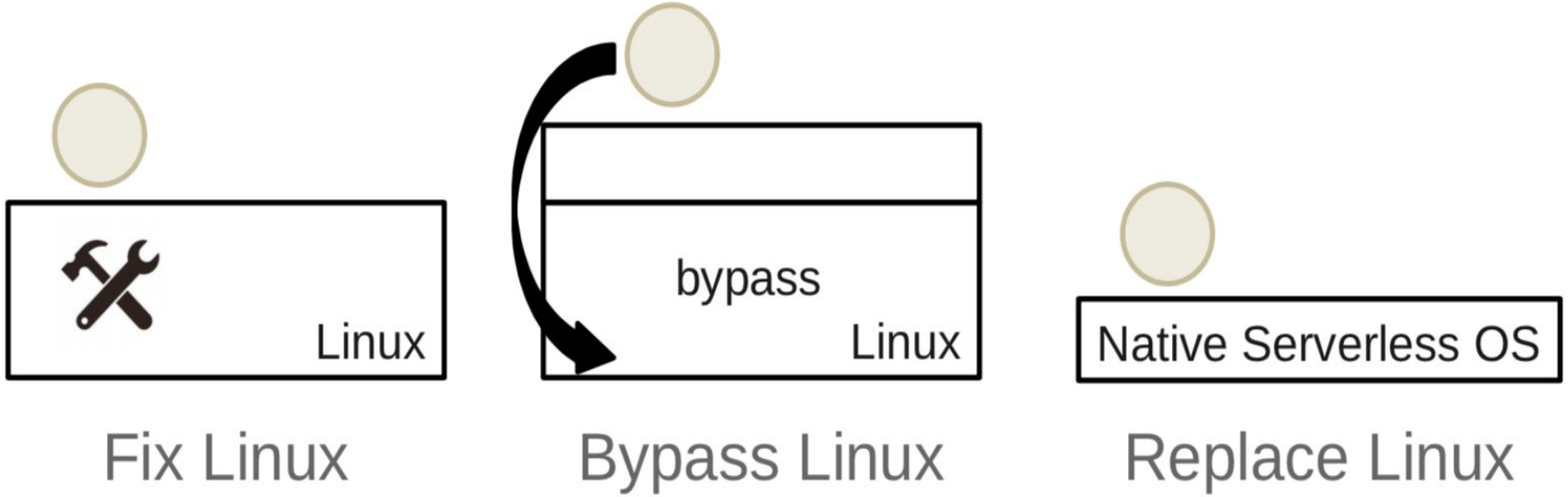


Precise definition varies from system to system.

Serverless Performance Requirements

- Latency
  - User: should launch immediately
  - Provider: low latency prevents caching complexities
  - Target: 100ms
- Throughput
  - Provider: should cover hourly cost of server
  - Target: 125 actions/sec.

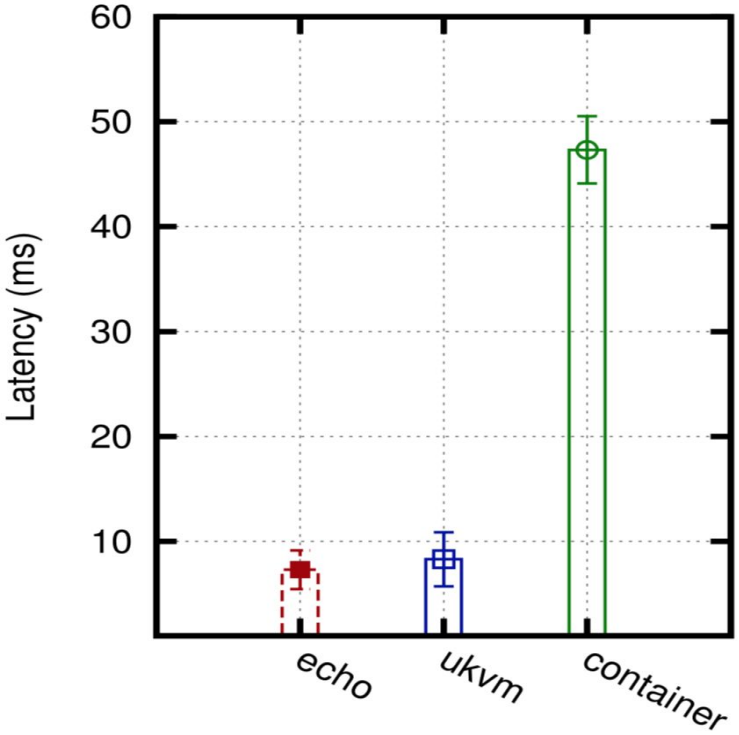
Options for Multi-tenant Implementation

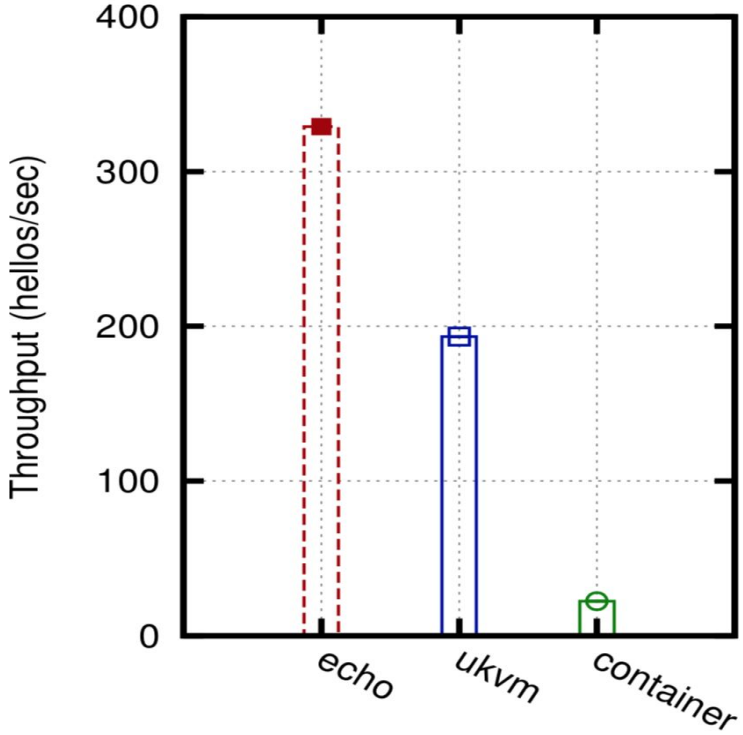


Experiment Setup

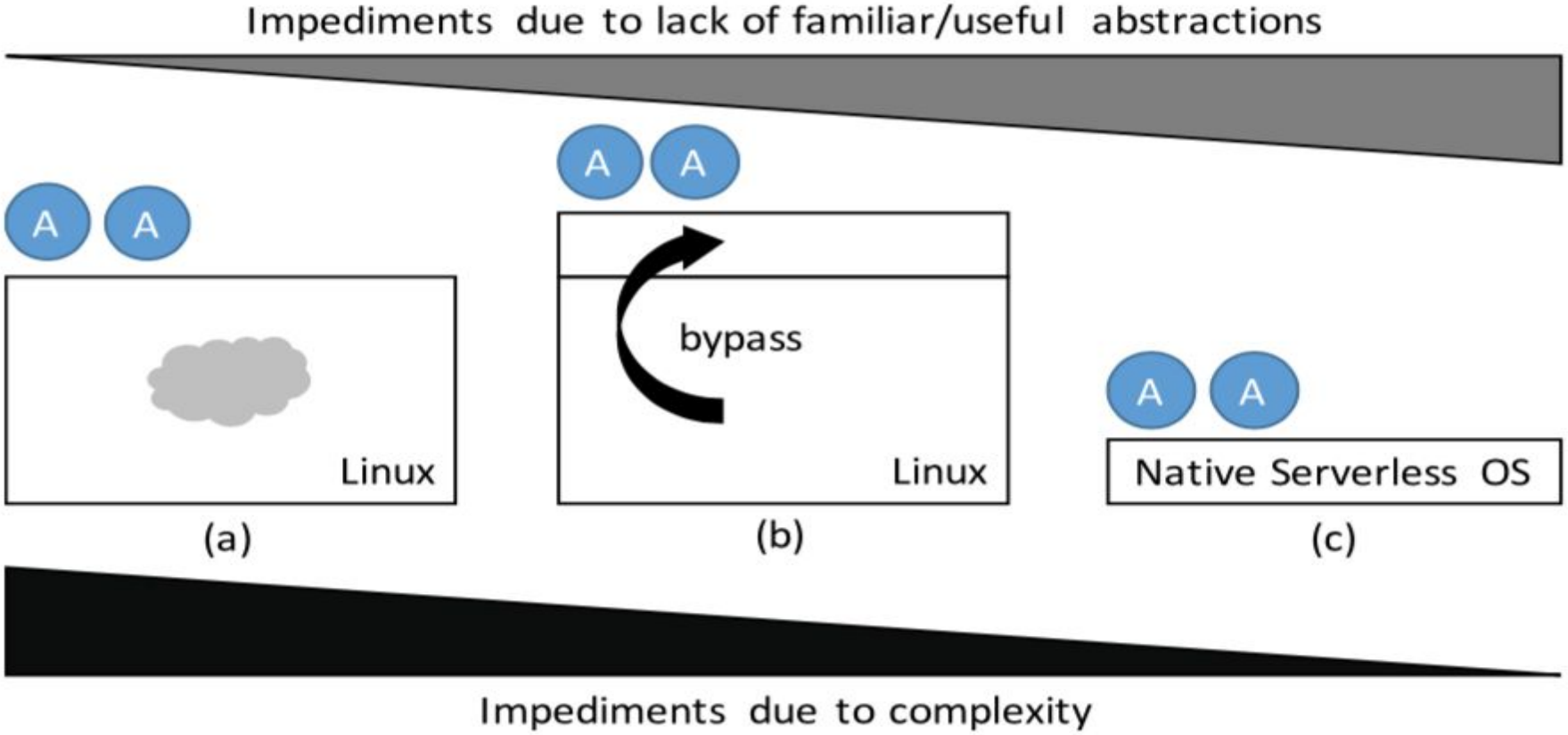
- Container
  - Reduce the complexity: microcontainer
- Unikernel
  - *Ukvm*
- Replace Linux
  - A raw LInux process *echo*

Experiment Results

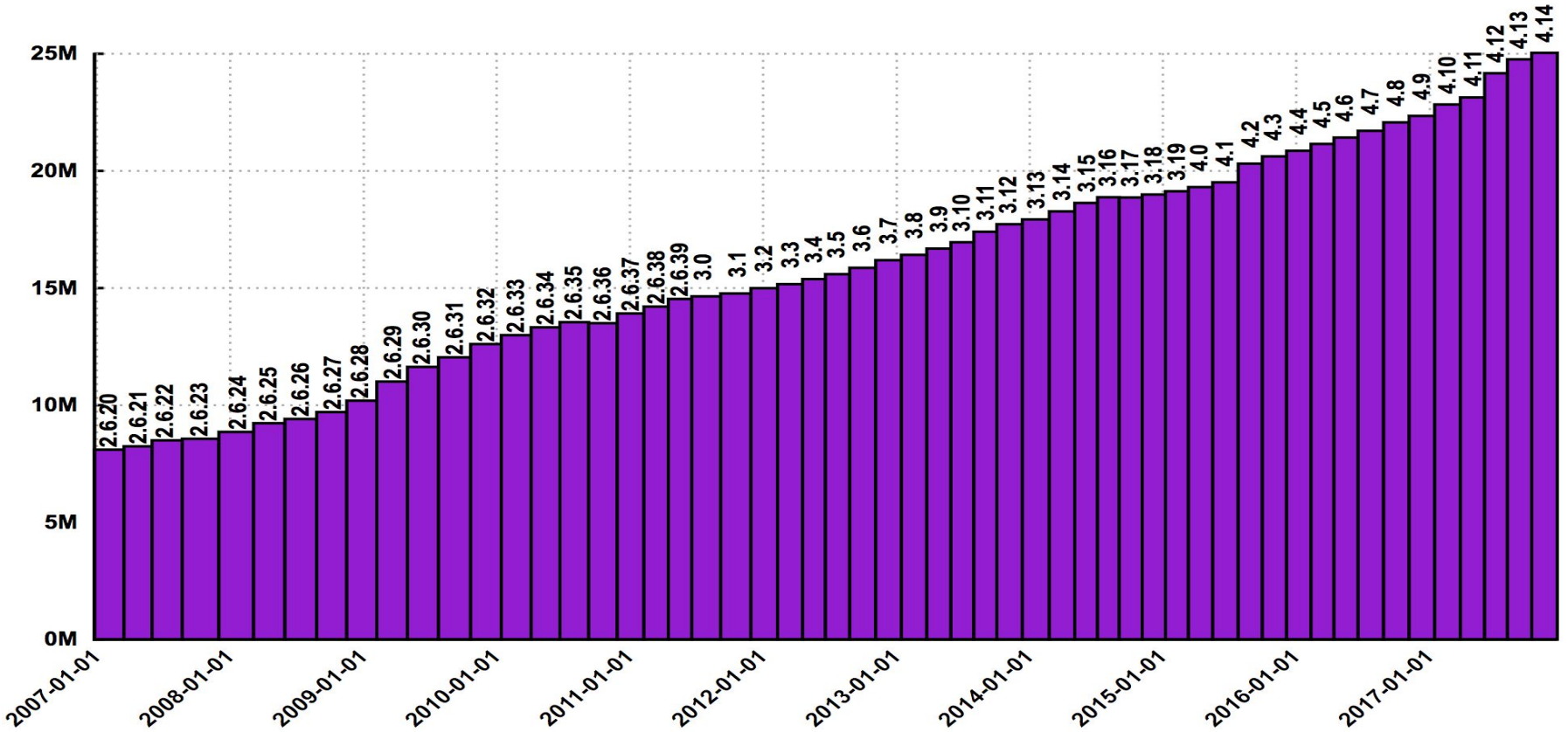




The Tradeoff



Fix Containers



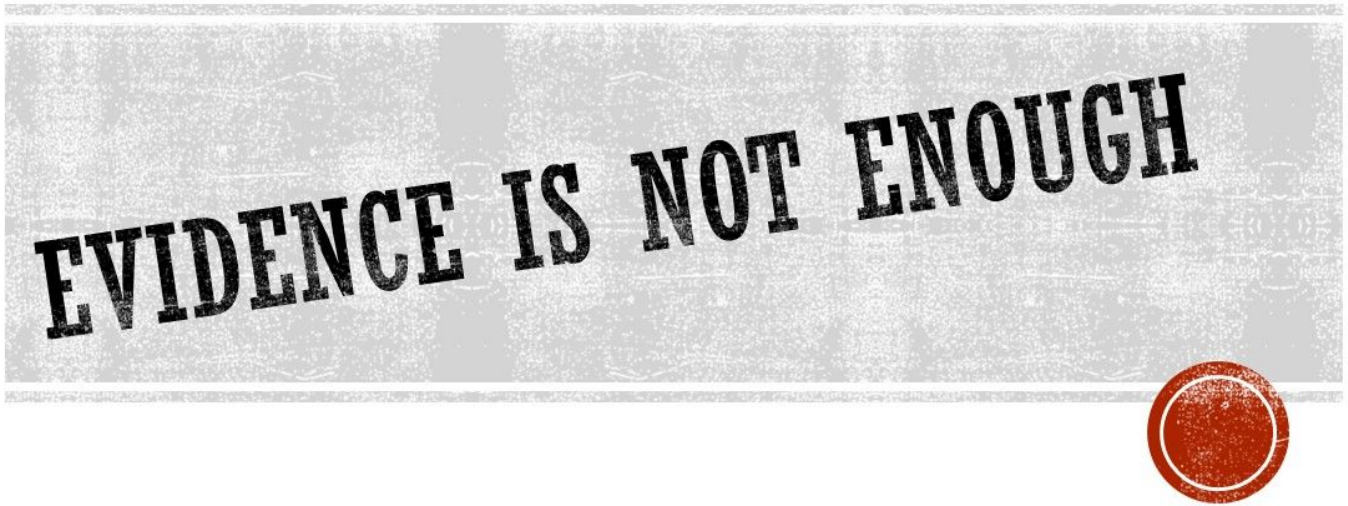
Bypassing Linux kernel

- A compromise for using container and rebuilding a host OS
- Easy way to speed up
- Less security
- Linux kernel isn’t designed for bypassing!

Replace Linux

- No Preemptible scheduling
  - Lambdas and actions are shorter to run and executed
- Don’t need to worry about synchronization a lot
- No IPC
- Limited set of I/O related calls

Common Questions



Conclusion

Native abstractions in Linux (e.g. containers) are not suitable for serverless architecture

Bypassing Linux kernel?

Replacing Linux entirely?