Analyzing the Serverless Cloud Virtualization Requirements

Student: Hong Fung Heng, Colin   Supervisors: Dr. Dmitrii Ustiugov & JooYoung Park

Current Serverless Hypervisors Suffer From Significant Virtualization Overhead

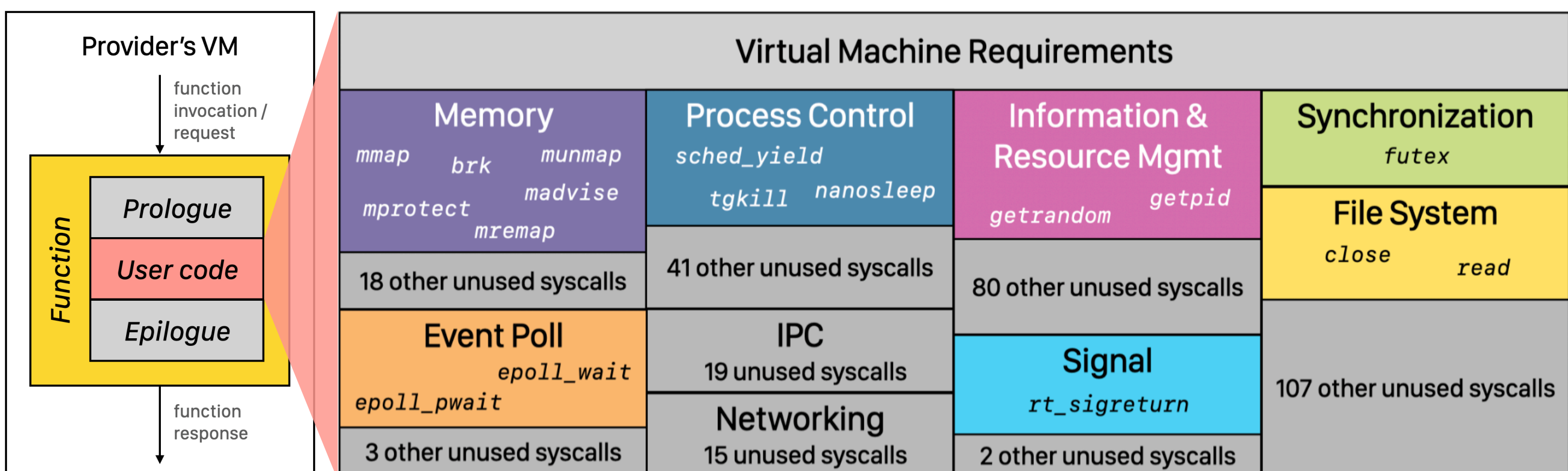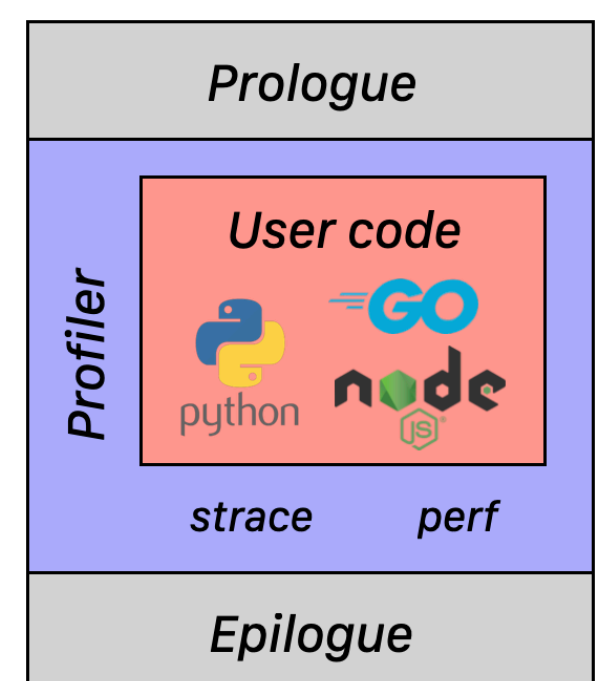Support for a vast number of unused syscalls leads to inefficient resource utilization

Mismatch between supported syscalls and actual user code requirements poses:

- A critical challenge in optimizing serverless environments

- Reduced cost-effectiveness and performance

Objective in our Investigation: Characterize Syscall Requirements

Profile user code of serverless workloads

Explore different language runtimes

Prologue

Profiler | User code (python, GO, node)

strace   perf

Epilogue

Provider's VM

function invocation / request

Function: Prologue / User code / Epilogue

function response

Virtual Machine Requirements

Memory
mmap   munmap
brk
mprotect   madvise
mremap
18 other unused syscalls

Process Control
sched_yield
tgkill   nanosleep
41 other unused syscalls

Information & Resource Mgmt
getrandom   getpid
80 other unused syscalls

Synchronization
futex

File System
close   read

Event Poll
epoll_wait
epoll_pwait
3 other unused syscalls

IPC
19 unused syscalls

Networking
15 unused syscalls

Signal
rt_sigreturn
2 other unused syscalls

107 other unused syscalls

Key Findings

Only 17 out of 303 Linux syscalls (<6%) required for user code execution across profiled functions

Specific syscall requirements vary by function and programming language

Opportunity to design lean hypervisors supporting only necessary syscalls per function

Future Work: Support Only What is Required

Insights for a serverless-native hypervisor design:

- Split prologue and epilogue from user code

- Modular / reconfigurable hypervisor architecture

- Tailor the hypervisor for user code's syscall API usage