

Storage Developer Conference September 22-23, 2020

Tracing and Visualizing FS Internals with SteBPF Superpowers

Suchakra Sharma ShiftLeft Inc.

Hani Nemati Microsoft

Suchakra Sharma

PhD, École Polytechnique de Montréal

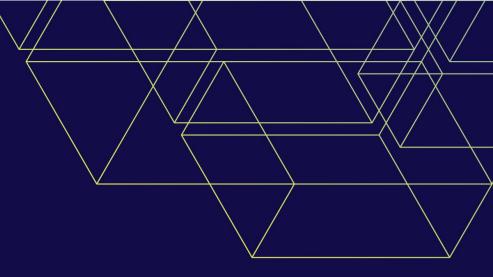
Staff Scientist
ShiftLeft Inc

Hani Nemati

PhD, École Polytechnique de Montréal

Software Engineer (Kernel)

Microsoft





Agenda

- FS Internals 101
- Journey of a read()
 - Tracing Control-Flow via ftrace
- FS and Performance
 - Probing the system via /proc
 - BCC and bpftrace Tool Collection
- Building tools with eBPF
 - Case Study: Read-ahead
 - DIY new BPF tool readaheadstat
 - Visualizing Performance



Why Filesystem?

Keep track of things

- What many files are open or being read?
- Which user/process is operating with files?
- How many files can be created?
- What data is most frequently accessed

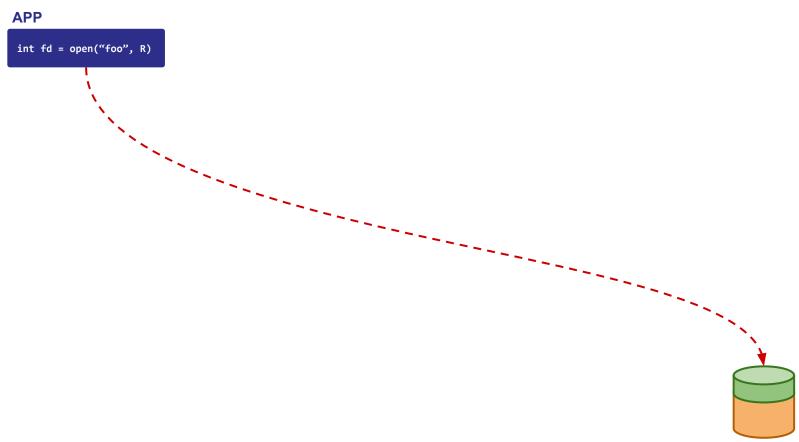
Structure

- How are they structured?
- What happens when we move them across?

Abstraction and Uniformity

- What if some of our files are on a network storage?
- What if we copy a file from two different storage devices?

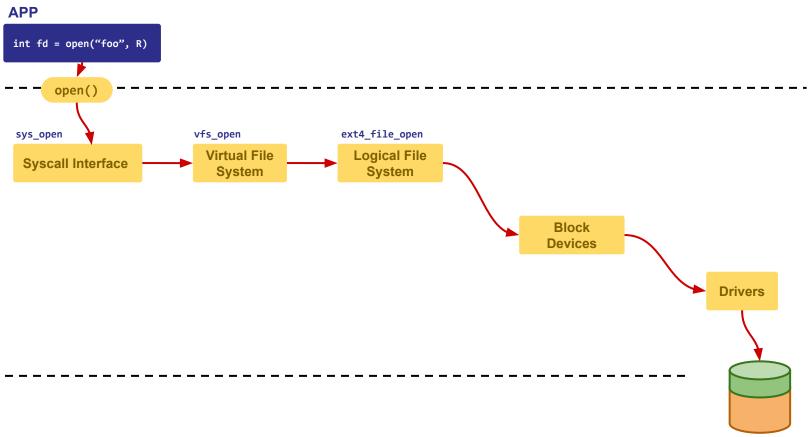




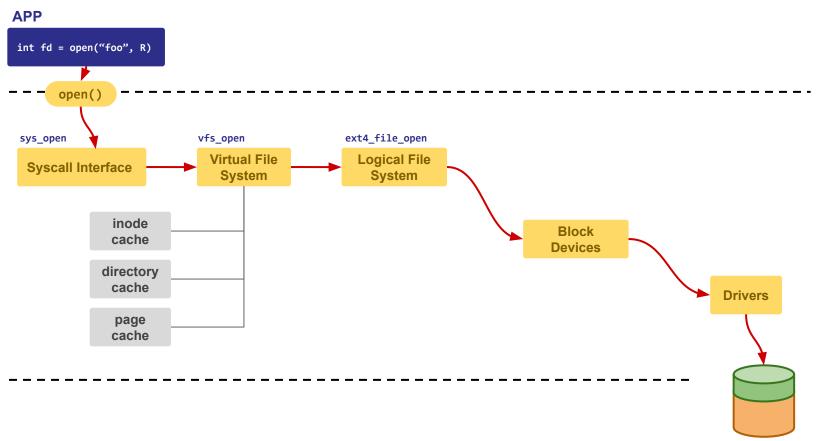


```
APP
int fd = open("foo", R)
       open()
                             vfs_open
                                                    ext4_file_open
 sys_open
                               Virtual File
                                                      Logical File
  Syscall Interface
                                System
                                                        System
                                                                                      Block
                                                                                     Devices
                                                                                                                Drivers
```



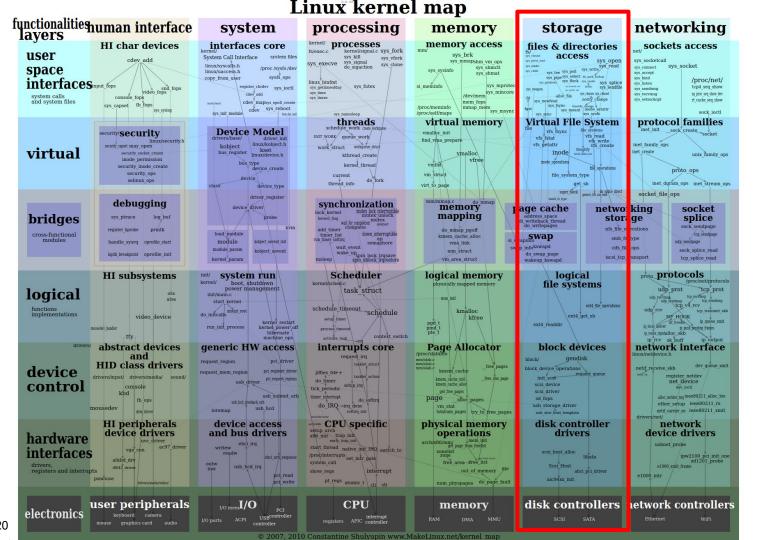


SD@



Linux kernel map functionalities numan interface layers system processing networking memory storage sockets access HI char devices kernel/ processes
fs/exec.c kernel/signal.c sys fork memory access files & directories interfaces core user sys_brk access System Call Interface system files sys kill sys vfork cdev add sys_execve sys_signal do sigaction sys_clone svs socketcall sys mmapshm vm ops sys tee sys poll sys write linux/syscalls.h sys_connect sys_socket sys shmctl space /proc /sysfs /dev linux/uaccess.h sys shmat sys tee or system sys spice sys spice sys flock sys roady sys splice system sys sys accept copy from user sys bind /proc/net/ interfaces input fops linux binfmt sys listen register_chrdev sys_ioctl sys mprotect sys gettimeofday sys futex tcp4 seq show svs sendmso sys mincore cdev add sg proc seg show dev system calls and system files sys recymsor console fops cdev mapsys epoll create mem fops rt cache seq show sys_capset fb_fops sys fsync tonouty hange y cdev sys reboot mmap_mem sys_msync /proc/meminfo sys systog sock joctl /proc/self/maps threads virtual memory **Virtual File System** protocol families inet init __sock_create __socket systems
vfs read
vfs write
vfs write
snotsly
serabout security/security **Device Model** vmalloc init INIT WORK queue work find vma prepare kobject linux/kobject.h vfs getattr work struct workqueue struct inet family ops security capset may open bus register kset linux/device.h virtual inet create security socket create vmalloc kthread create unix family ops inode permission bus type device create kernel thread vmlist security inode create proto ops security ops vm struct file system type device selinux ops do fork thread info inet_dgram_ops inet stream ops device type super block generic file ain read socket file ops driver register debugging memory do mmap synchronization device driver page cache networking socket mapping bridges log buf wat in onnistin mutex add timer complete owner splice storage sock sendpage nfs file operations register kprobe printk cross-functional modules load module si swapinio top sendpage kmem cache alloc smb_fs_type handle_sysrq oprofile_start module kobiect nevent init udp sendpage vma link wait event swap infokswapd cifs file ops module param sock splice read wake_up_spin_lock_irqsave spin_unlock_irqrestore mm struct do swap page kodb breakpoint oprofile init iscsi_tcp_transport tcp splice read kernel param vm area struct wakeup kswapd Scheduler logical proto protocols HI subsystems init/ system run logical memory boot shutdown kernel/sched c file systems physically mapped memory udp prot tcp prot power management task struct logical udp recymag tcp recymsg tcp sendmsg tcp v4 rcv mm init start kernel schedule timeout schedule ext4 file operations functions implementations kmalloc NF HOOK ip queue xmit video device setup timer kfree kernel restart ext4 readdir run init process mousedev handler ip push pending frames activate task context switch ip rcv sk buff ip output Page Allocator abstract devices generic HW access interrupts core block devices network interface and request irq **HID class drivers** block device operations dev queue xmit netif receive skb device pci register driver kmem cache init scst drivers/input/ drivers/media/ sound/ kmem cache init free one page pci request regions usb driver do_timer net device tick_periodic setup_irq control kmem cache alloc scsi device dev ioctl get free pages scsi driver kbd alloc netdev mq leee80211 alloc hw usb submit urb page alloc pages sd fops fb ops do softirg ash had diveback urb ether setup jeee80211 rx do IRQ irq desc usb storage driver usb hcd netif carrier on ieee80211 xmit drm driver totalram pages try to free pages CPU specific HI peripherals device access physical memory disk controller network device drivers and bus drivers operations drivers device drivers hardware x86 init trap init arch/x86/mm/ mem init vga con ac97 driver start thread native init IRQ switch to interfaces scsi host alloc /proc/interrupts set intr gate ipw2100 pci init one zd1201 probe ehci urb enqueue atkbd dry free area free list i8042 driver usb hcd irg Scsi Host ahci pci driver out of memory die registers and interrupts show regs pci read e1000 intr aic94xx init pt regs atomic t cli sti pci write num physpages do page fault user peripherals **CPU** disk controllers network controllers memory electronics keyboard camera I/O ports ACPI USB^{controller} registers APIC controller mouse graphics card audio







- \$ sudo trace-cmd record -p function -P 2535
- \$ sudo trace-cmd report



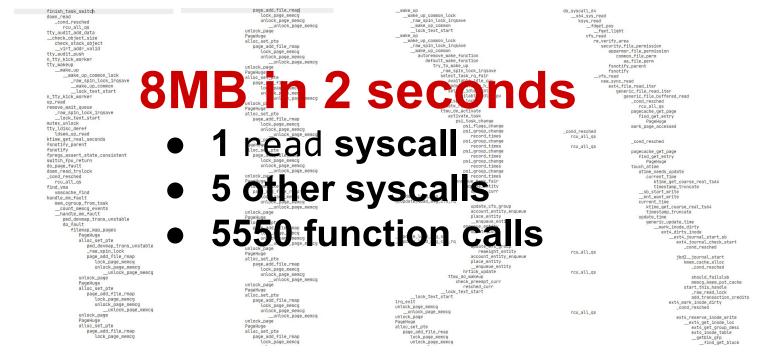
- sudo trace-cmd record -p function -P 2535
- \$ sudo trace-cmd report

```
page_add_file_rmap
finish task switch
                                                                                            lock_page_mencg
              _cond_resched
                                                                                                _unlock_page_memcg
                 rcu_all_qs
           tty_audit_add_data
                                                                                      PageHuge
            check object size
                                                                                      alloc set nte
             check_stack_object
                                                                                        page add file rmap
              virt addr valid
                                                                                           Lock page mencg
           tty audit nush
                                                                                           unlock_page_memcg
           n_ttv_k1ck_worker
                                                                                               __unlock_page_memcg
           ttv wakeup
              __wake_up
                                                                                      PageHuge
                 __wake_up_common_lock
                                                                                      alloc set pte
                    _raw_spin_lock_irqsave
                                                                                        page_add_file_rmap
                     wake up common
                                                                                           lock page menco
                     lock_text_start
                                                                                           unlock page memor
           n tty kick worker
                                                                                                _unlock_page_memcq
           un read
           remove_wait_queue
                                                                                      PageHuge
             _raw_spin_lock_irqsave
                                                                                      alloc_set_pte
               _lock_text_start
                                                                                        page_add_file_rmap
           mutex_unlock
                                                                                           lock page memog
           tty_ldisc_deref
                                                                                           unlack page nemco
              ldsem_up_read
                                                                                                _unlock_page_memcg
           ktime_get_real_seconds
                                                                                      unlock_page
           fsnotify_parent
                                                                                      PageHuge
           fsnotify
           fpregs_assert_state_consistent
                                                                                        page_add_file_rmap
           switch fou return
                                                                                           lock page memog
           do_page_fault
                                                                                           unlock page memog
                                                                                               _unlock_page_memcg
           down_read_trylock
                                                                                      unlock_page
           _cond_resched
              rcu_all_qs
                                                                                      PageHuge
                                                                                      alloc_set_pte
           find_vma
                                                                                        page_add_file_rmap
              ymacache find
                                                                                           lock_page_mencg
           handle_mm_fault
                                                                                            unlock_page_memcg
             mem_cgroup_from task
               count memcg events
                                                                                               _unlock_page_memcg
                                                                                      unlock_page
              handle mm fault
                 nmd devman trans unstable
                                                                                      PageHuge
                                                                                      alloc set pte
                 do fault
                                                                                        page_add_file_rmap
                    filemap_map_pages
                                                                                            lock_page_memcg
                       alloc set pte
                                                                                                _unlock_page_memcg
                          pmd_devmap_trans_unstable
                                                                                      unlock_page
                          _raw_spin_lock
                                                                                      PageHuge
                          page_add_file_rmap
                                                                                      alloc set ote
                             lock_page_memcg
                                                                                        page_add_file_rmap
                             unlock_page_memcg
                                                                                            lock_page_memcq
                                 __unlock_page_memcg
                                                                                            unlock_page_memcg
                                                                                               _unlock_page_memcg
                       PageHuge
                                                                                      unlock_page
                       alloc_set_pte
                                                                                      PageHuge
                          page_add_file_rmap
                                                                                      alloc set nte
                             lock_page_memcg
                                                                                        page_add_file_rmap
                             unlock_page_memcg
                                                                                           lock page memog
                                 __unlock_page_memcg
                                                                                            unlock_page_memcg
                       unlock_page
                                                                                               _unlock_page_memcg
                       PageHuge
                       alloc set ote
                          page_add_file_rmap
                                                                                      alloc set nte
                                                                                        page_add_file_rmap
                             lock page memog
                                                                                           lock_page_memcg
```

```
do_syscall_64
  _wake_up_common_lock
                                                                     x64 sys read
     _raw_spin_lock_irqsave
                                                                        ksys_read
       _wake_up_common
                                                                          __fdget_pos
      _lock_text_start
                                                                               _fget_light
                                                                           vfs_read
  __wake_up_common_lock
     _raw_spin_lock_irqsave
      __wake_up_common
        autoremove wake function
           default wake function
              try_to_wake_up
                  raw_spin_lock_irqsave
                  select_task_rq_fair
                    available idle cou
                    update_cfs_rq_h_load
                       available_idle_cpu
                  raw spin lock
                  undate ro clock
                  ttwu do activate
                    activate_task
                       psi_task_change
                          psi_flags_change
                          nst group change
                                                                  cond resched
                             record_times
                                                                    rcu all as
                          psi_group_change
                             record times
                                                                     rcu all os
                          nst group change
                             record times
                          psi_group_change
                             record_times
                          ps1_group_change
                             record times
                       enqueue_task_fair
                          enqueue_entity
                             update_curr
undate_load_avq_se
_update_load_avg_cfs_rg
                             update_cfs_group
                             place_entity
                              _enqueue_entity
                          enqueue_entity
                             update_curr
update load avg se
update load avg cfs ro
                             update cfs group
                               reweight entity
                                                                     rcu_all_qs
                             place_entity
                         __enqueue_entity
                                                                     rcu_all_qs
                    ttwu do wakeup
                       check_preempt_curr
                          resched curr
                  lock text start
       _lock_text_start
unlock_page_memcg
    _unlock_page_memcg
                                                                     rcu all os
unlock page
PageHuge
alloc_set_pte
  page_add_file_rmap
     lock_page_memcg
     unlock_page_memcg
```

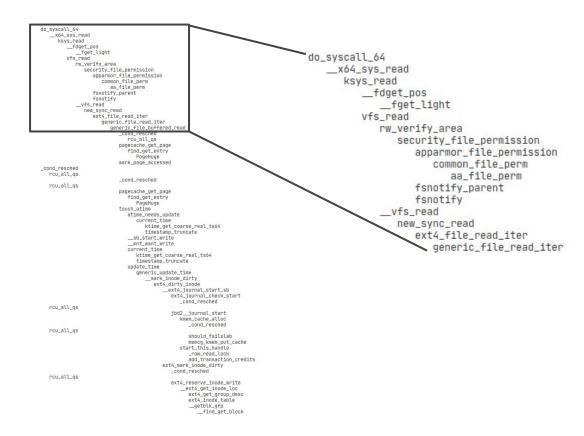
```
rw_verify_area
   security_file_permission
      apparmor_file_permission
         common_file_perm
            as file nerm
      fsnotify_parent
      fsnotify
   new_sync_read
      ext4 file read iter
         generic_file_read_iter
            generic_file_buffered_read
               _cond_resched
                  rcu_all_gs
               namerache met name
                  find_get_entry
               mark_page_accessed
                cond resched
               pagecache_get_page
                  find get entry
                     PageHuge
                touch_atime
                  atime_needs_update
                     current_time
                        ktime_get_coarse_real_ts64
                        timestamp truncate
                   _sb_start_write
                    _mnt_want_write
                   current time
                     ktime get coarse real ts64
                      timestamp truncate
                   update_time
                     generic_update_time
                        __mark_inode_dirty
                           ext4 dirty inode
                               _ext4_journal_start_sb
                                 ext4_journal_check_start
                                 1hd2 inurnal start
                                    kmem_cache_alloc
                                        should fallslah
                                       memca kmem nut cache
                                    start_this_handle
                                        add_transaction_credits
                              ext4 mark_inode_dirty
                                  _cond_resched
                                 ext4 reserve inode write
                                    __ext4_get_inode_loc
                                       ext4_get_group_desc
ext4_inode_table
                                        __getblk_gfp
                                           __find_get_block
```

- sudo trace-cmd record -p function -P 2535
- \$ sudo trace-cmd report

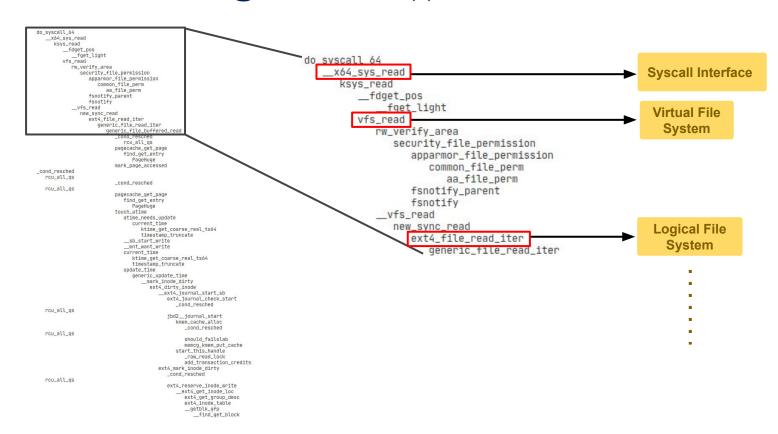


```
__x64_sys_read
     ksys_read
        __fdget_pos
            _fget_light
           rw_verify_area
             security_file_permission
                apparmor_file_permission
                   common_file_perm
                      aa_file_perm
                 fsnotify_parent
                 fsnotify
           __vfs_read
              new_sync_read
                ext4_file_read_iter
                   generic_file_read_iter
                      generic_file_buffered_read
                            rcu_all_qs
                          pagecache_get_page
                            find_get_entry
                               PageHuge
                          mark_page_accessed
_cond_resched
  rcu_all_qs
                          _cond_resched
  rcu_all_qs
                          pagecache_get_page
                            find get entry
                               PageHuge
                          touch_atime
                            atime_needs_update
                               current time
                                  ktime get coarse real ts64
                                  timestamp_truncate
                             _sb_start_write
                             __mnt_want_write
                               ktime get coarse real ts64
                               timestamp_truncate
                             undate time
                               generic_update_time
                                   mark inode dirty
                                     ext4_dirty_inode
                                        __ext4_journal_start_sb
                                           ext4_journal_check_start
                                              _cond_resched
  rcu_all_qs
                                           ibd2__iournal_start
                                              kmem_cache_alloc
                                                 _cond_resched
  rcu_all_qs
                                                 should failslab
                                                 memcq_kmem_put_cache
                                              start_this_handle
                                                 _raw_read_lock
                                                 add transaction credits
                                        ext4 mark inode dirty
                                           _cond_resched
  rcu_all_qs
                                           ext4_reserve_inode_write
                                               __ext4_get_inode_loc
                                                 ext4_get_group_desc
                                                 ext4_inode_table
                                                 __getblk_gfp
                                                    __find_get_block
```







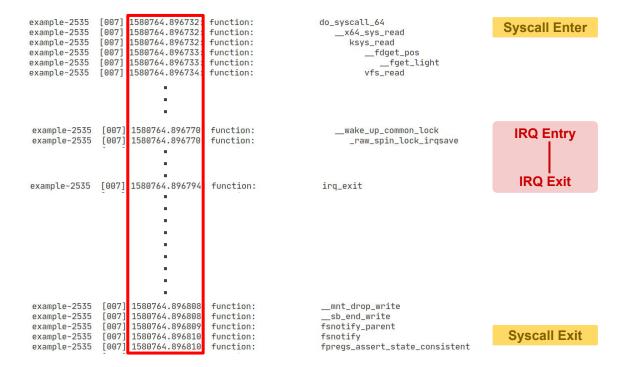




```
example-2535 [007] 1580764.896732: function:
                                                          do_syscall_64
example-2535 [007] 1580764.896732: function:
                                                             __x64_sys_read
example-2535 [007] 1580764.896732: function:
                                                                ksys_read
example-2535 [007] 1580764.896733: function:
                                                                   __fdget_pos
example-2535 [007] 1580764.896733: function:
                                                                      __fget_light
example-2535 [007] 1580764.896734: function:
                                                                   vfs read
              [007] 1580764.896770: function:
                                                             __wake_up_common_lock
example-2535
                                                                _raw_spin_lock_irgsave
example-2535
              [007] 1580764.896770: function:
             [007] 1580764.896794: function:
                                                          irg_exit
example-2535 [007] 1580764.896808: function:
                                                          mnt drop write
example-2535 [007] 1580764.896808: function:
                                                          __sb_end_write
example-2535 [007] 1580764.896809: function:
                                                          fsnotify_parent
example-2535 [007] 1580764.896810: function:
                                                          fsnotify
example-2535 [007] 1580764.896810: function:
                                                          fpregs_assert_state_consistent
```



```
example-2535
                    1580764.896732
                                    function:
                                                           do_syscall_64
example-2535
              [007] 1580764.896732
                                    function:
                                                              __x64_sys_read
example-2535
                    1580764.896732
                                     function:
                                                                 ksys_read
example-2535
              [007] 1580764.896733:
                                    function:
                                                                     __fdget_pos
example-2535
              [007] 1580764.896733:
                                    function:
                                                                       __fget_light
example-2535
              [007] 1580764.896734; function:
                                                                    vfs read
example-2535
                    1580764.896770
                                     function:
                                                              __wake_up_common_lock
example-2535
              [007]
                    1580764.896770
                                     function:
                                                                 _raw_spin_lock_irgsave
              [007] 1580764.896794
                                     function:
                                                           ira_exit
example-2535
example-2535
              [007] 1580764.896808
                                     function:
                                                            mnt drop write
example-2535
               [007]
                    1580764.896808
                                     function:
                                                            __sb_end_write
example-2535
              [007]
                    1580764.896809
                                     function:
                                                            fsnotify_parent
example-2535
              [007]
                    1580764.896810
                                     function:
                                                            fsnotify
example-2535
              [007] 1580764.896810
                                     function:
                                                            fpregs_assert_state_consistent
```





FS and Performance



What can /proc tell us?

cat /proc/diskstats







What can /proc tell us?

```
259
         0 nvme0n1 1823879 1141588 58777929 328845 3512978 2244953 151315050
7397322 0 2759684 7859940 88819 0 543261592 2631 273795 107459
          1 nvme0n1p1 4443 0 27072 2273 2 0 2 34 0 392 2310 4 0 1034184 1 0 0
 259
          2 nvme0n1p2 573 398 59794 77 367 115 151352 415 0 776 525 91 0
259
645136 33 0 0
          3 nvme0n1p3 1820929 1141190 58667503 326372 3238845 2244838
259
151163696 7287293 0 2747016 7639943 88724 0 541582272 2627 0 0
                                                     Milliseconds spent in
                                                     all disk reads
         Milliseconds spent in
          all 10s
```

cat /proc/diskstats

¹https://www.kernel.org/doc/Documentation/iostats.txt

What can /proc tell us?

cat /proc/1475/io

rchar: 723810543

wchar: 1609216447277

syscr: 27156292 syscw: 142054696

read_bytes: 30699520

write bytes: 6516736

cancelled write bytes: 5160960

Number of bytes requested to be read from block device by this

process



SD@

Building tools over /proc and /sys

```
# iostat -d nvme0n1
Linux 5.7.1-050701-generic (isengard)
                                        09/11/2020
                                                      x86 64
                                                                  (8 CPU)
Device
                    kB_read/s kB_wrtn/s
                                           kB_read
                                                      kB wrtn
              tps
              7.61
                        42.62
                                  109.59
                                           30521808
                                                     78476133
nvme0n1
                                     Total kilobytes written
                                     per second
```

iotop
(Exercise left for the reader)

Targeted Analysis

```
example-2535
              [007] 1580764.896732: function:
                                                         do syscall 64
                                                                                              Syscall Enter
example-2535
             [007] 1580764.896732: function:
                                                             __x64_sys_read
example-2535 [007] 1580764.896732: function:
                                                               ksys_read
example-2535 [007] 1580764.896733: function:
                                                                  __fdget_pos
                                                                     __fget_light
example-2535 [007] 1580764.896733: function:
example-2535 [007] 1580764.896734: function:
                                                                  vfs read
              [007] 1580764.896770: function:
                                                            wake up common lock
example-2535
                                                                                                IRQ Entry
              [007] 1580764.896770: function:
                                                                _raw_spin_lock_irgsave
example-2535
                                                                                                 IRQ Exit
             [007] 1580764.896794: function:
example-2535
                                                          irg_exit
example-2535
              [007] 1580764.896808: function:
                                                          __mnt_drop_write
example-2535
              [007] 1580764.896808: function:
                                                          sb end write
example-2535
              [007] 1580764.896809: function:
                                                          fsnotify_parent
                                                                                               Syscall Exit
              [007] 1580764.896810: function:
example-2535
                                                          fsnotify
              [007] 1580764.896810: function:
                                                          fpregs_assert_state_consistent
example-2535
```

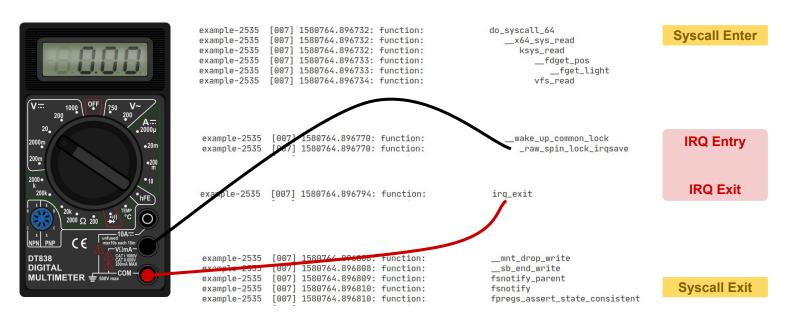






SD@

Targeted Analysis

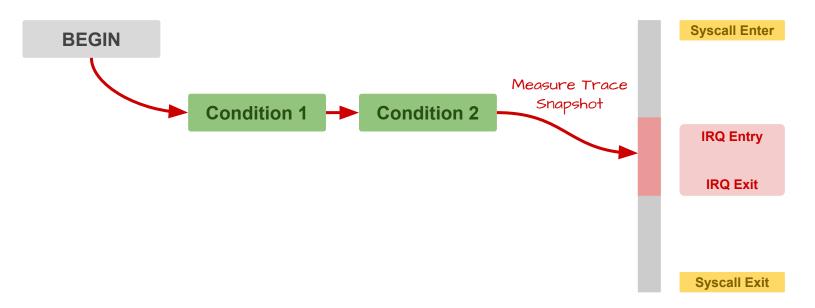


¹https://github.com/iovisor/bpftrace/tree/master/tools



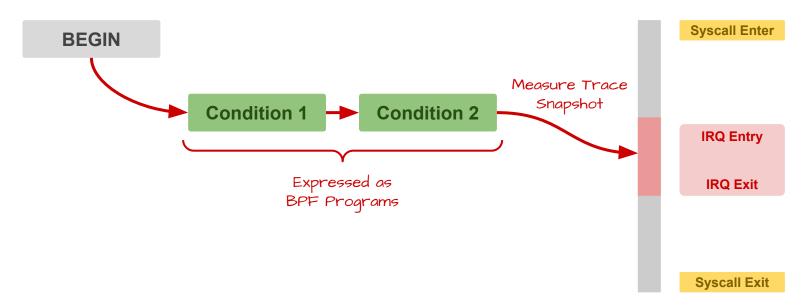


Live and Programmatic Analysis





Live and Programmatic Analysis



Tracing with BPF

bpftrace Tools

```
Most block operations
# ./biolatency.bt
                                                                                                                                                                                                                                                                                                                                                          took between 1024 to
Attaching 3 probes...
                                                                                                                                                                                                                                                                                                                                                         2048 micro-seconds
 Tracing block device I/O... Hit Ctrl-C to end.
 ^(
@usecs:
  [256, 512)
  [512, 1K)
                                                                                                                                      10
  [1K, 2K)
                                                                                                                                     426
                                                                                                                                                                  [2K, 4K)
                                                                                                                                                               oxedsymbol{eta} തെരുതെ തെരുത്തെ തെരുത്ത്തെ തെരുത്തെ തെരുത്ത്രെ തെരുത്തെ തെരുത്ത
  [4K, 8K)
  [8K, 16K)
                                                                                                                                                                128
  [16K, 32K)
                                                                                                                                     68
                                                                                                                                                                  | @@@@@@@@
 [32K, 64K)
  [64K, 128K)
  [128K, 256K)
                                                                                                                                      10
                                                                                                                                                                |@
```

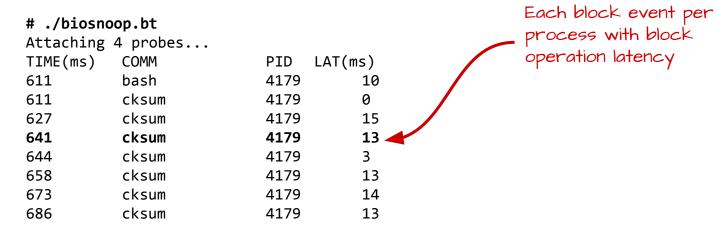


¹https://github.com/iovisor/bpftrace/tree/master/tools

Tracing with BPF

SD@

bpftrace Tools



¹https://github.com/iovisor/bpftrace/tree/master/tools







bpftrace Tools

biostack.bt
bitesize.bt
dcsnoop.bt
mdflush.bt

BCC Tools

biostat.py
bitesize.py
bitehist.py
biotop.py

Building tools with





eBPF for Observability

Probe the OS, Apps, Network

- Single programmatic way of tracing and monitoring each and every aspect of the software infrastructure
- Hook to any userspace or kernel function* or predefined tracepoints, then extract and visualize data
- Enable and disable probes, live filtering, low runtime overhead

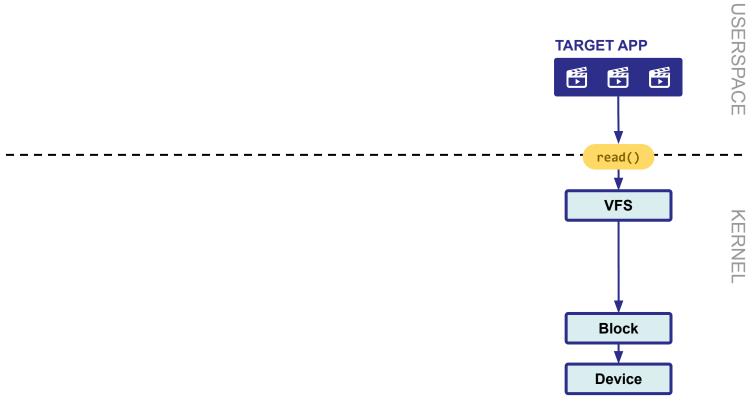
Workflow

- Create your observation tool in userspace
- Compile it and send it to kernel and attach to hooks
- It runs and returns data back via buffer or maps

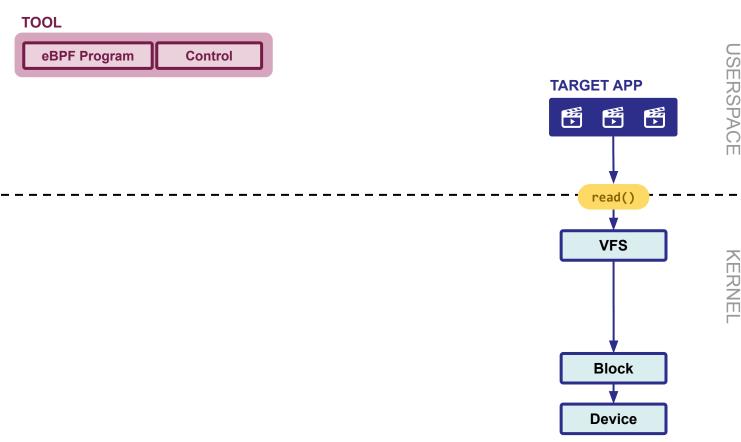


eBPF for Observability

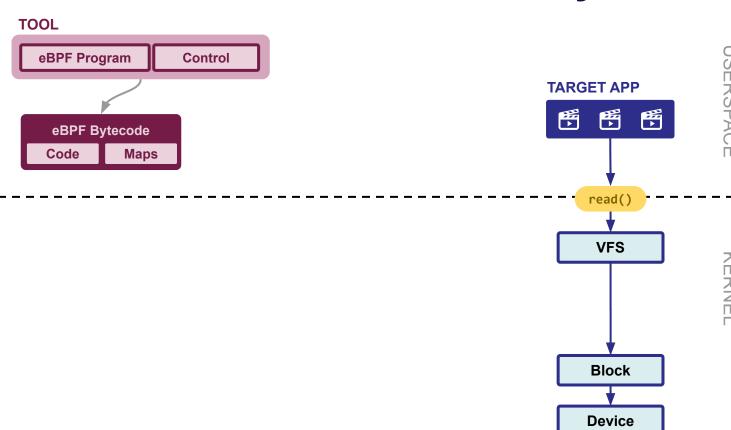




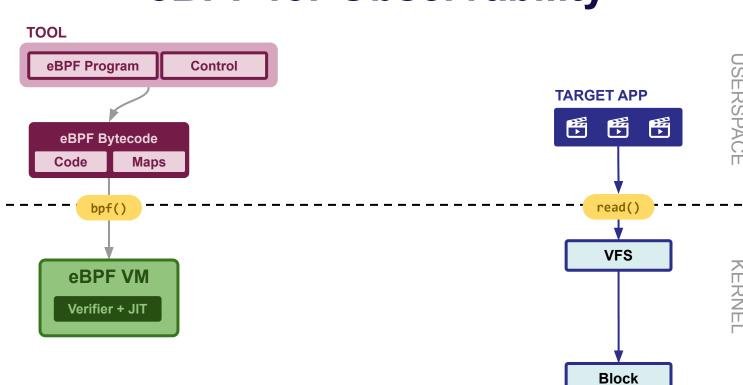
eBPF for Observability







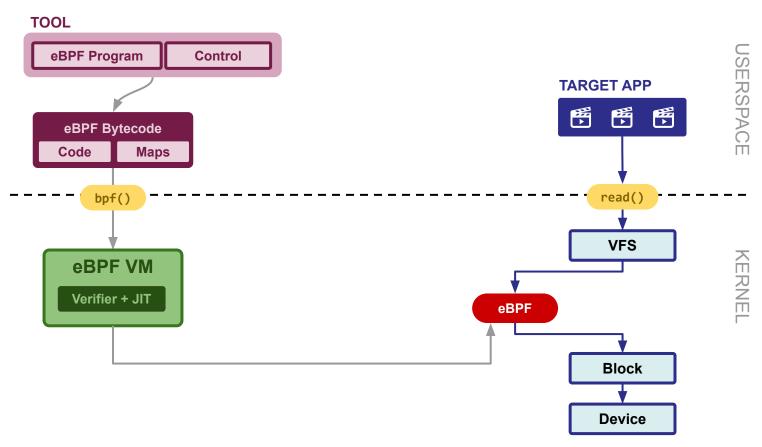




Device

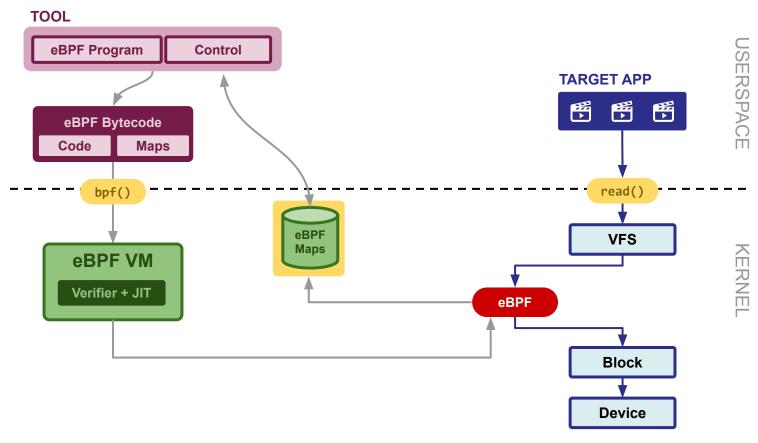




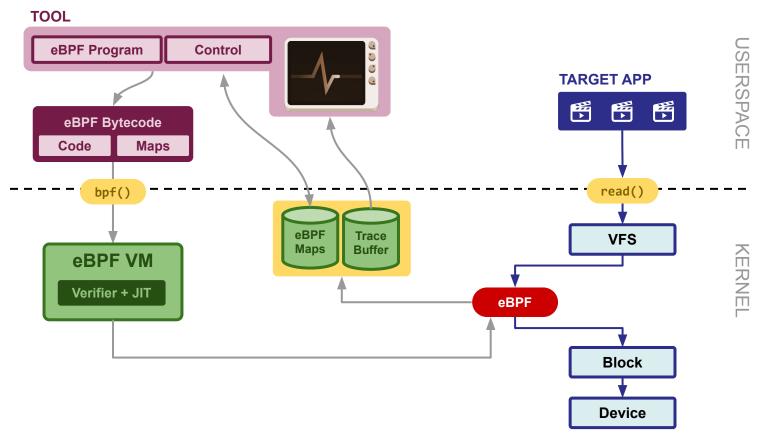




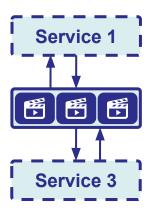




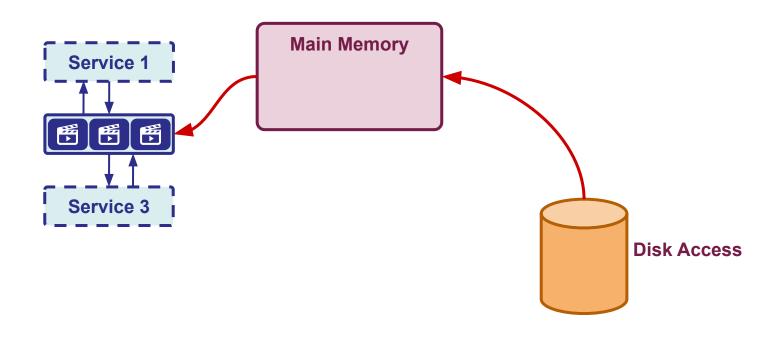




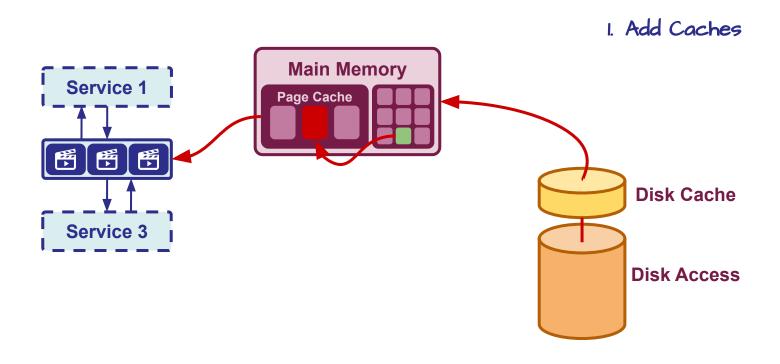


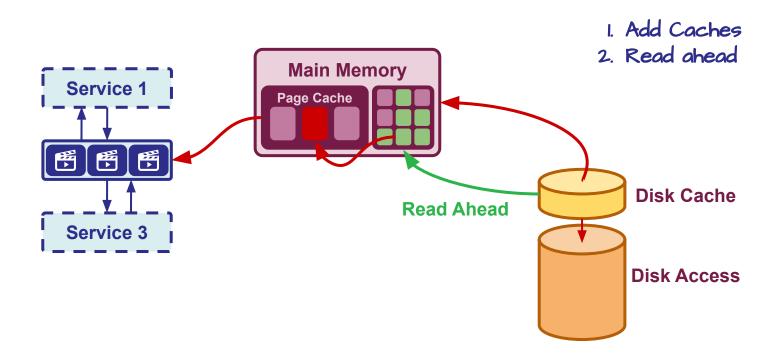






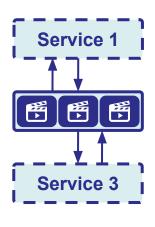






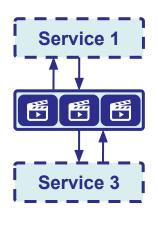






How?

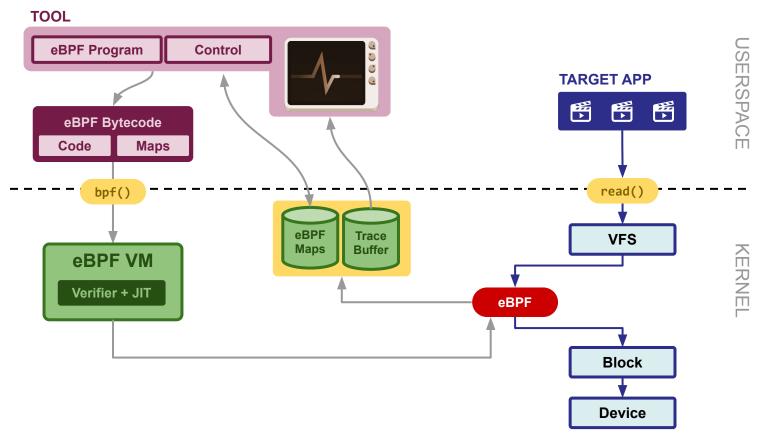
- Apps can "advise" the OS to use efficient read-ahead mechanism for sequential reads (like the one here)
 - madvise (addr, len, MADV_SEQUENTIAL)
- OS enables an aggressive read-ahead of pages in memory since it now know faults will be less due to seg read.



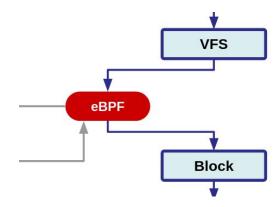
Problems

- Do apps really know how to use it?
- Modern apps are unaware of underlying infrastructure
- Lack of insight on how the different hardwares/OS optimize this really
- Has been a lack of dynamic and deep-observability tools to "see" the problems in real time



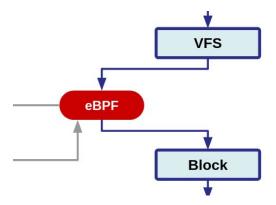








- Where and what are the probes?
 - Dynamic:
 - Kprobes/Kretprobes (Kernel)
 - USDT/Uprobes (Userspace)
 - Static: Tracepoints

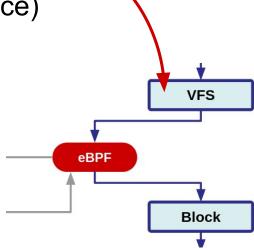




- Where and what are the probes?
 - Dynamic:

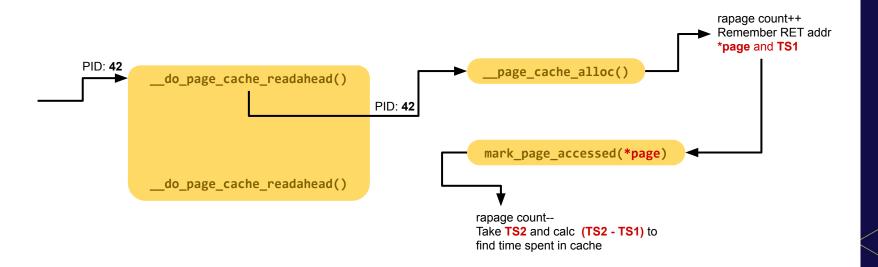
Kprobes/Kretprobes (Kernel) -USDT/Uprobes (Userspace)

• Static: Tracepoints



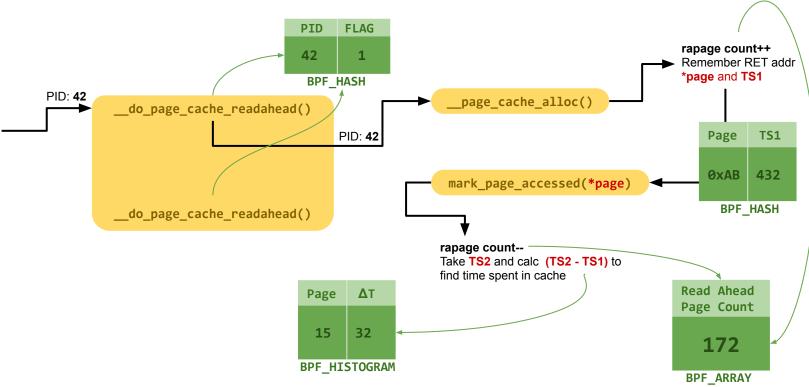


Selecting Kprobes/Kretprobes





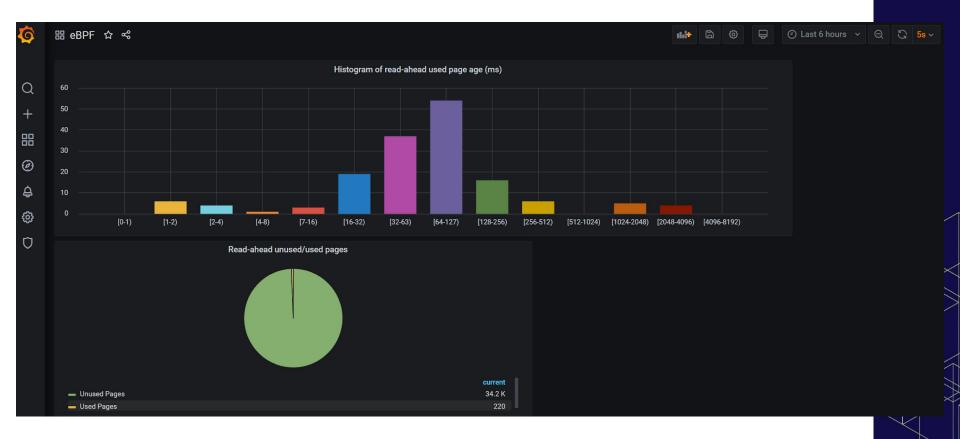
Maintaining state and storing data in hash-maps













Or e-mail us suchakra@shiftleft.io hanemati@microsoft.com