THE LINUX FOUNDATION
OPEN SOURCE SUMMIT

When eBPF Meets FUSE

Improving the performance of user file systems

Ashish Bijlani, PhD Student, Georgia Tech

@ashishbijlani

THE LINUX FOUNDATION

In-Kernel vs User File Systems

*"People who think that userspace filesystems are realistic for anything but toys are just **misguided**."*

- Linus Torvalds

"A lot of people once thought Linux and the machines it ran on were toys…

Apparently I'm misguided."

- Jeff Darcy

In-Kernel vs User File Systems

- **Examples**
  - Ext4, OverlayFS, etc.
- **Pros**
  - Native performance
- **Cons**
  - Poor security/reliability
  - Not easy to develop/debug/maintain

- **Examples**
  - EncFS, Gluster, etc.
- **Pros**
  - Improved security/reliability
  - Easy to develop/debug/maintain
- **Cons**
  - **Poor performance!**

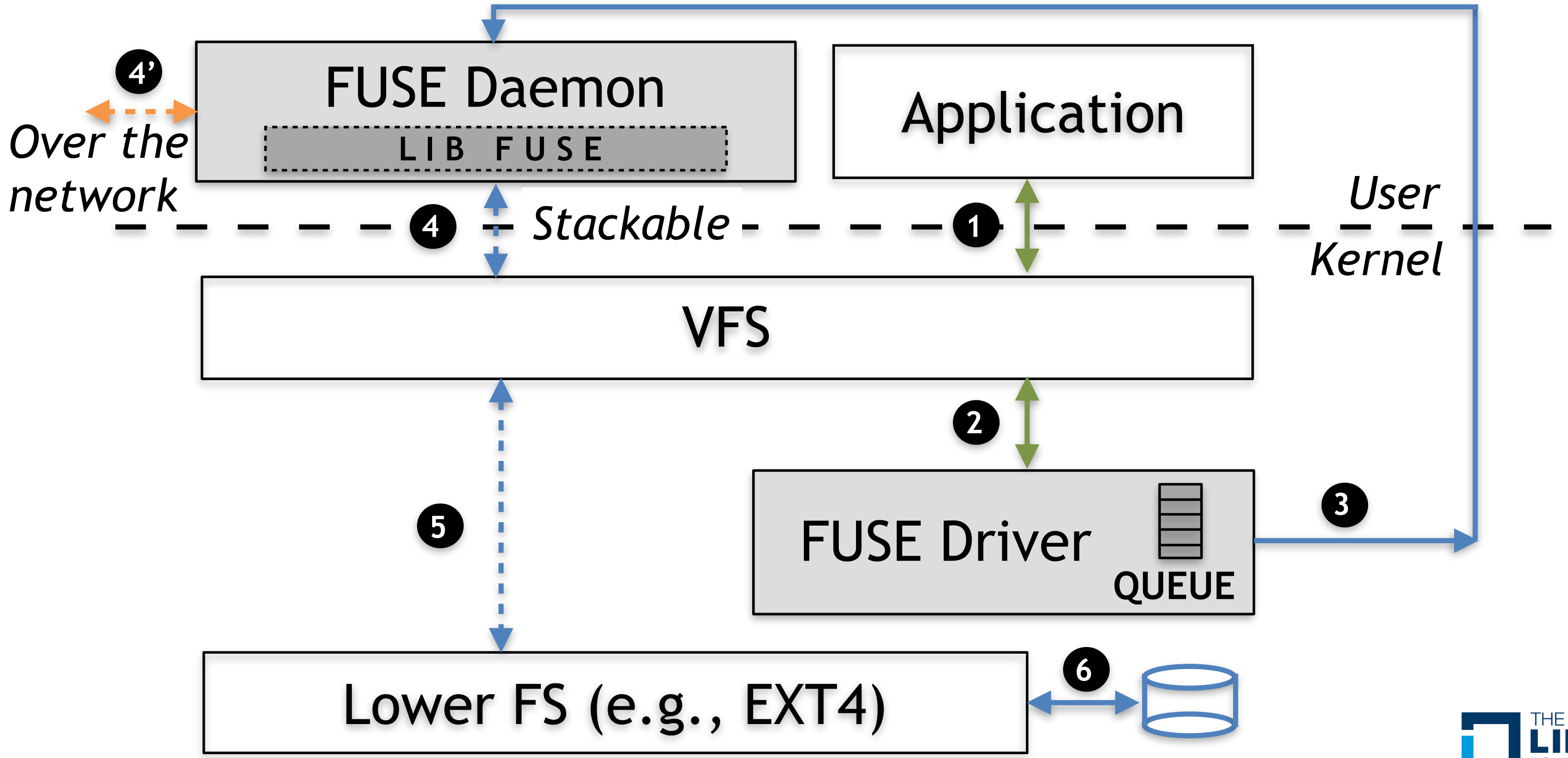
THE LINUX FOUNDATION

File Systems in User Space (FUSE)

- State-of-the-art framework
  - All file system handlers implemented in user space
- Over 100+ FUSE file systems
  - Stackable: Android SDCardFS, EncFS, etc.
  - Network: GlusterFS, Ceph, Amazon S3FS, etc.

```
struct fuse_lowlevel_ops ops {
        .lookup  = handle_lookup,
        .access  = NULL,
        .getattr = handle_getattr,
        .setattr = handle_setattr,
        .open    = handle_open,
        .read    = handle_read,
        .readdir = handle_readdir,
        .write   = handle_write,
        // more handlers …
        .getxattr = handle_getxattr,
        .rename = handle_rename,
        .symlink = handle_symlink,
        .flush   = NULL,
}
```

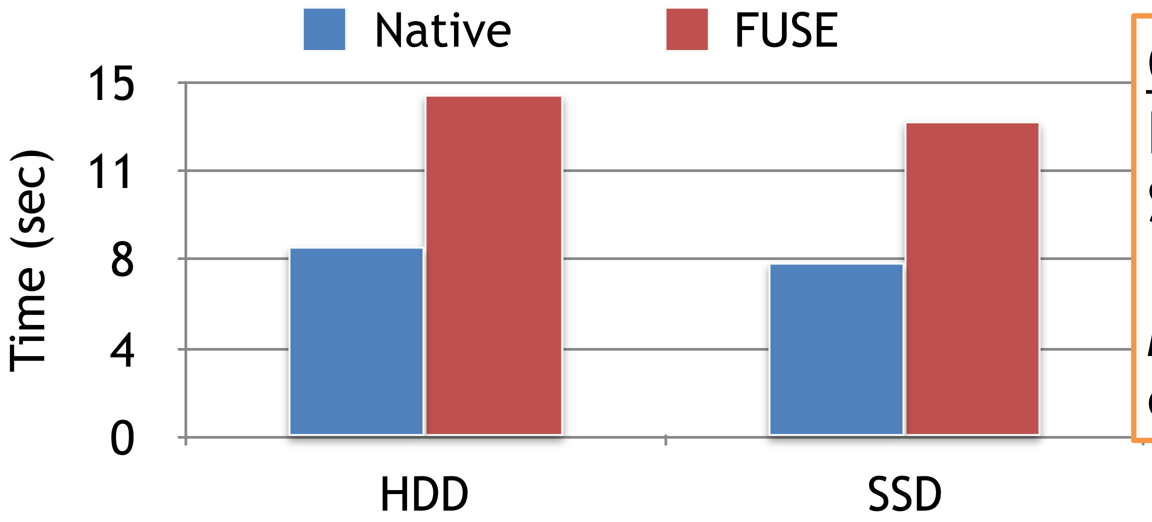
FUSE Architecture



THE LINUX FOUNDATION

FUSE Performance

- ***"tar xf linux-4.17.tar.xz"***
  - Intel i5-3350 quad core, Ubuntu 16.04.4 LTS
  - Linux 4.11.0, LibFUSE commit # 386b1b



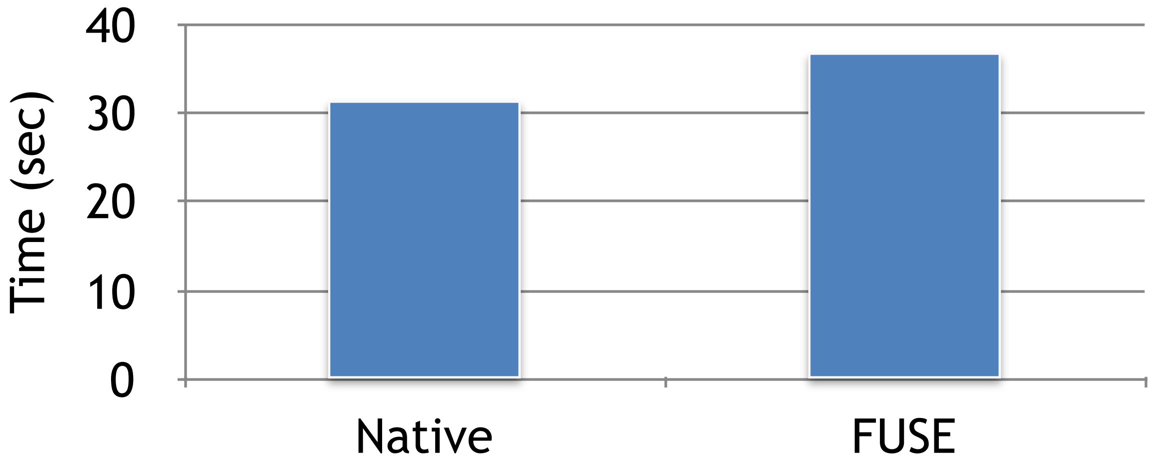
<u>Overhead</u>
HDD: 78.8%
SSD: 81.1%

More noticeable on faster media!

THE LINUX FOUNDATION

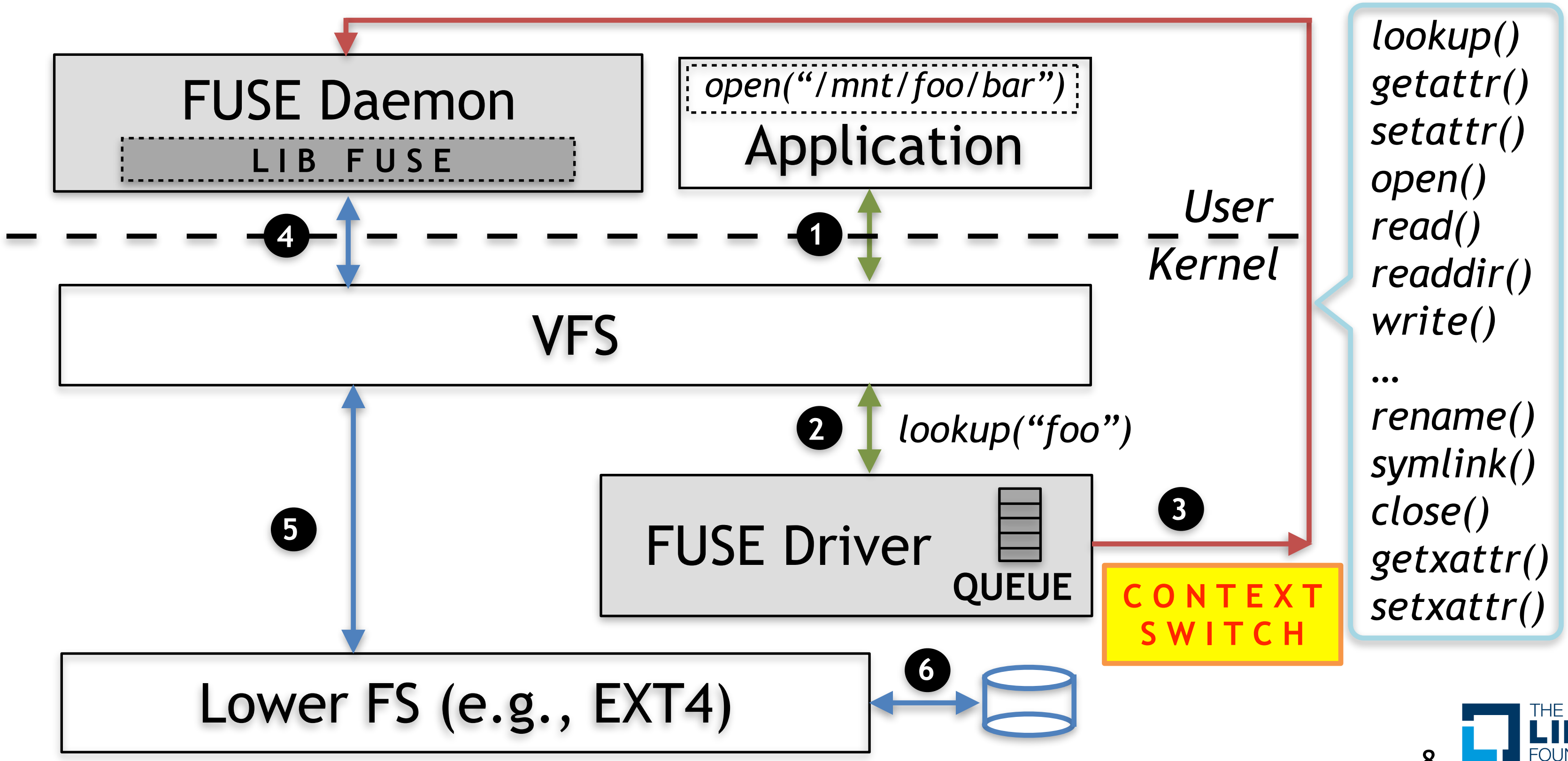
FUSE Performance

- ***"cd linux-4.17; make tinyconfig; make -j4"***
  - Intel i5-3350 quad core, SSD, Ubuntu 16.04.4 LTS
  - Linux 4.11.0, LibFUSE commit # 386b1b



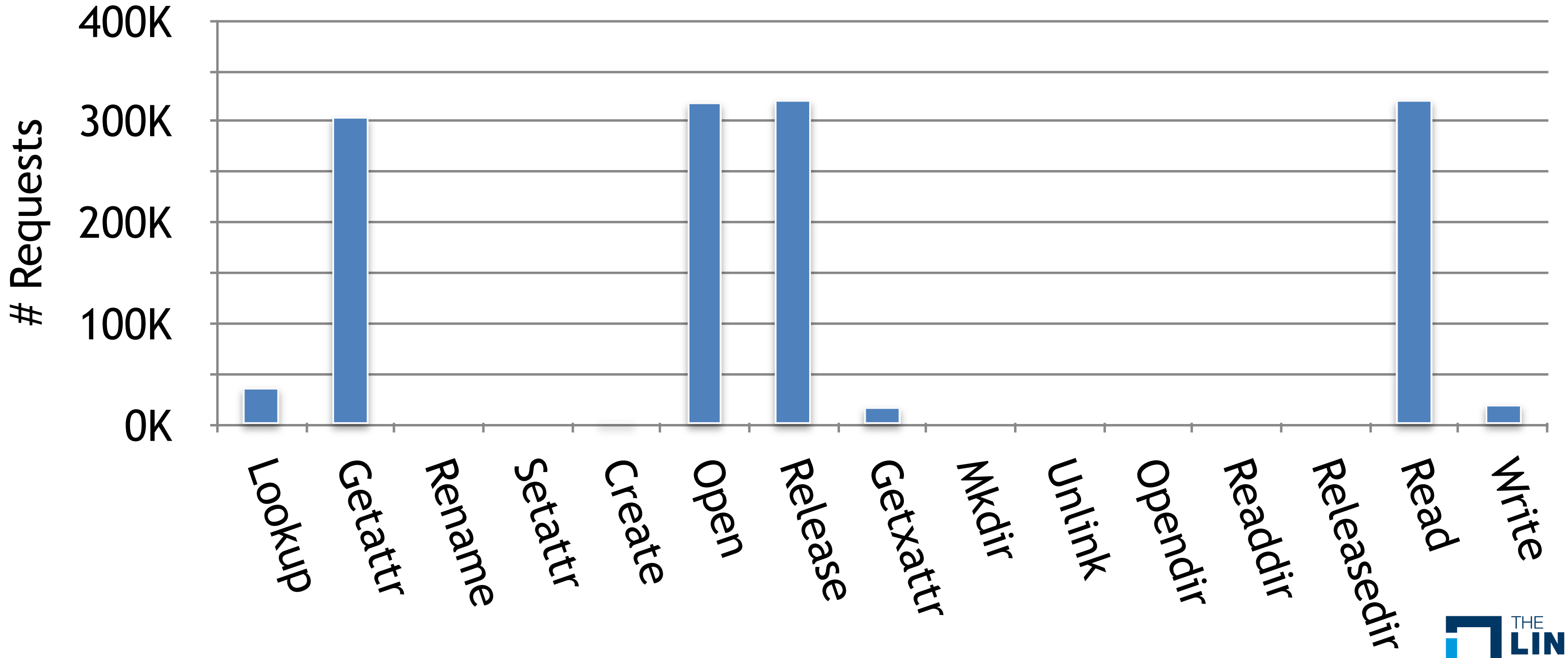
17.54% overhead

FUSE Architecture



Requests received by FUSE daemon

- ***"cd linux-4.17; make tinyconfig; make -j4"***



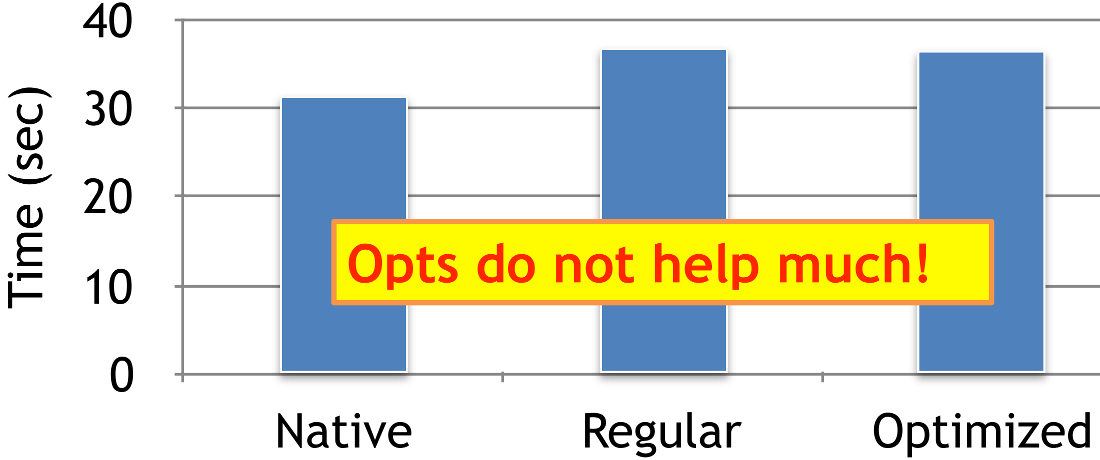
THE LINUX FOUNDATION

FUSE Optimizations

- Big 128K writes
  - *"-o max_write=131072"*
- Zero data copying for data I/O
  - *"-o splice_read, splice_write, splice_move"*
- Leveraging VFS caches
  - Page cache for data I/O
    - *"-o writeback_cache"*
  - Dentry and Inode caches for lookup() and getattr()
    - *"entry_timeout", "attr_timeout"*

FUSE Performance

- ***"cd linux-4.17; make tinyconfig; make -j4"***
  - Intel i5-3350 quad core, Ubuntu 16.04.4 LTS
  - Linux 4.11.0, LibFUSE commit # 386b1b

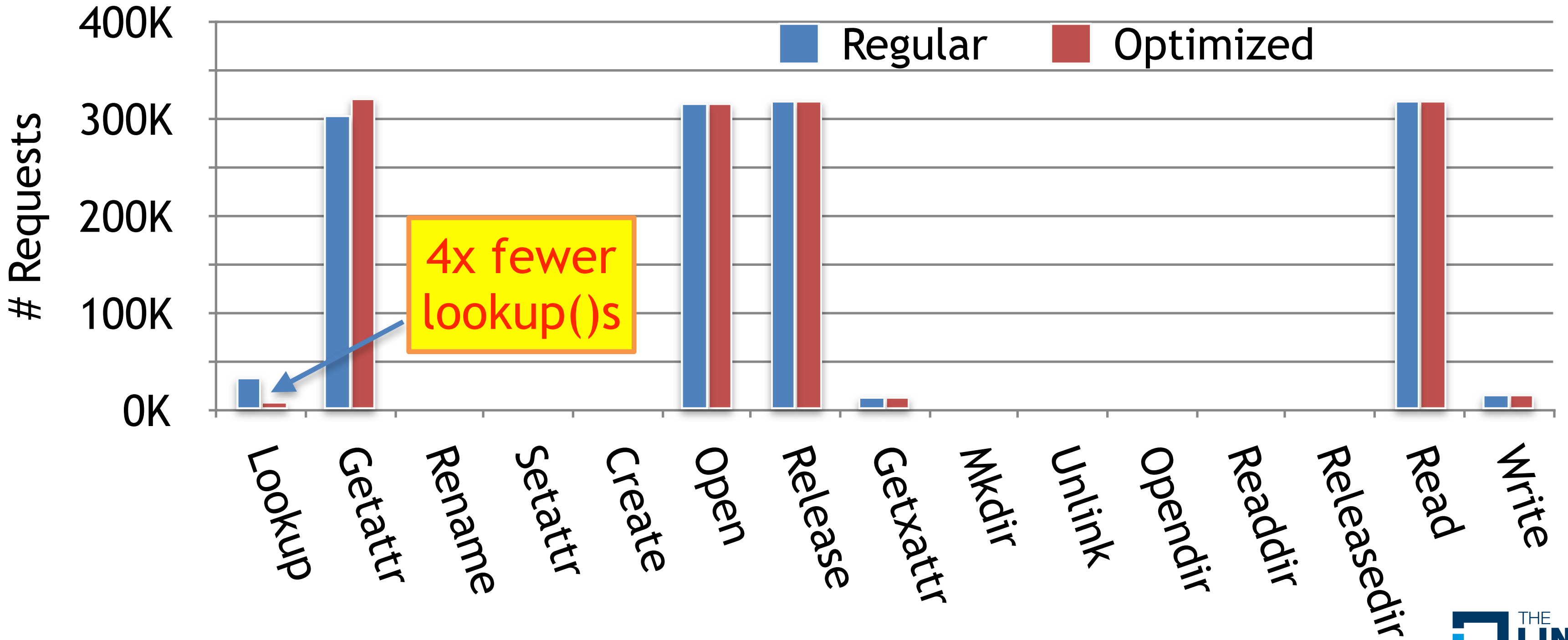


<u>Opts Enabled</u>

```
-o max_write=128K
-o splice_read
-o splice_write
-o splice_move
entry_timeout > 0
attr_timeout > 0
```

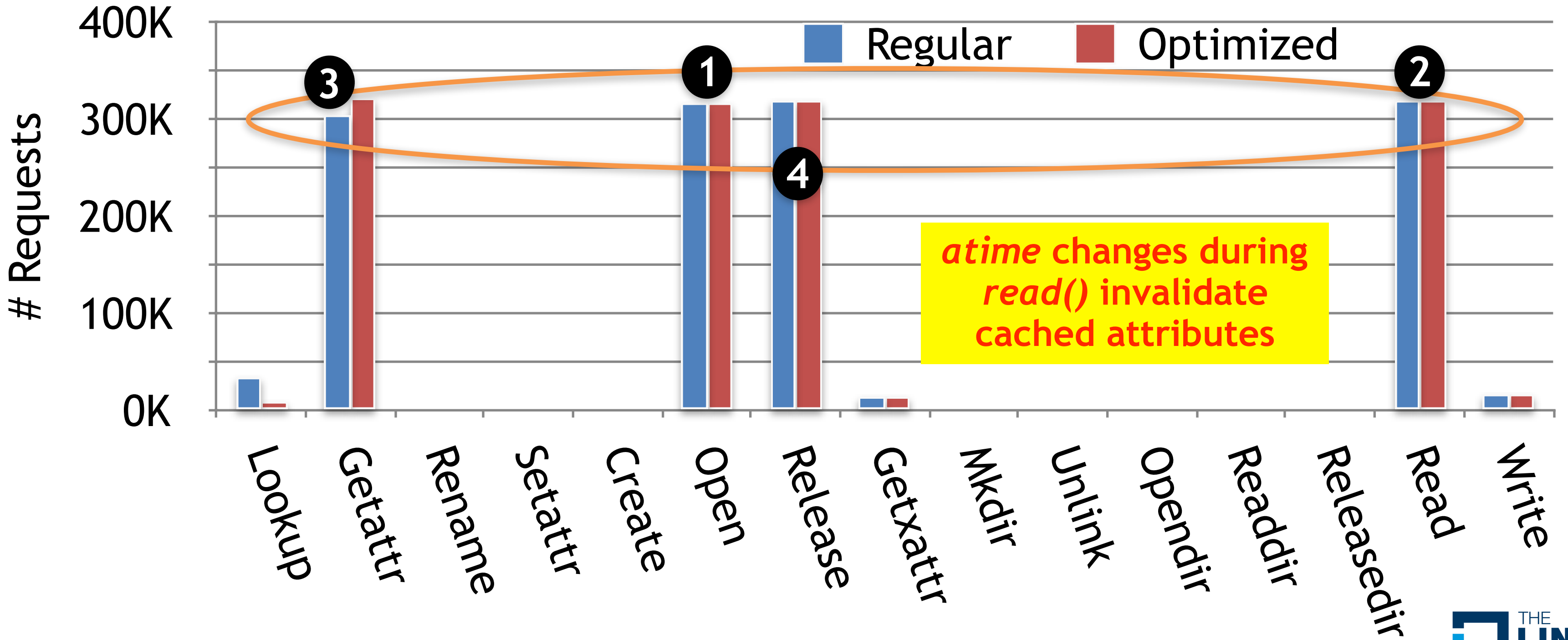
Requests received by FUSE daemon

- ***"cd linux-4.17; make tinyconfig; make -j4"***



Requests received by FUSE daemon

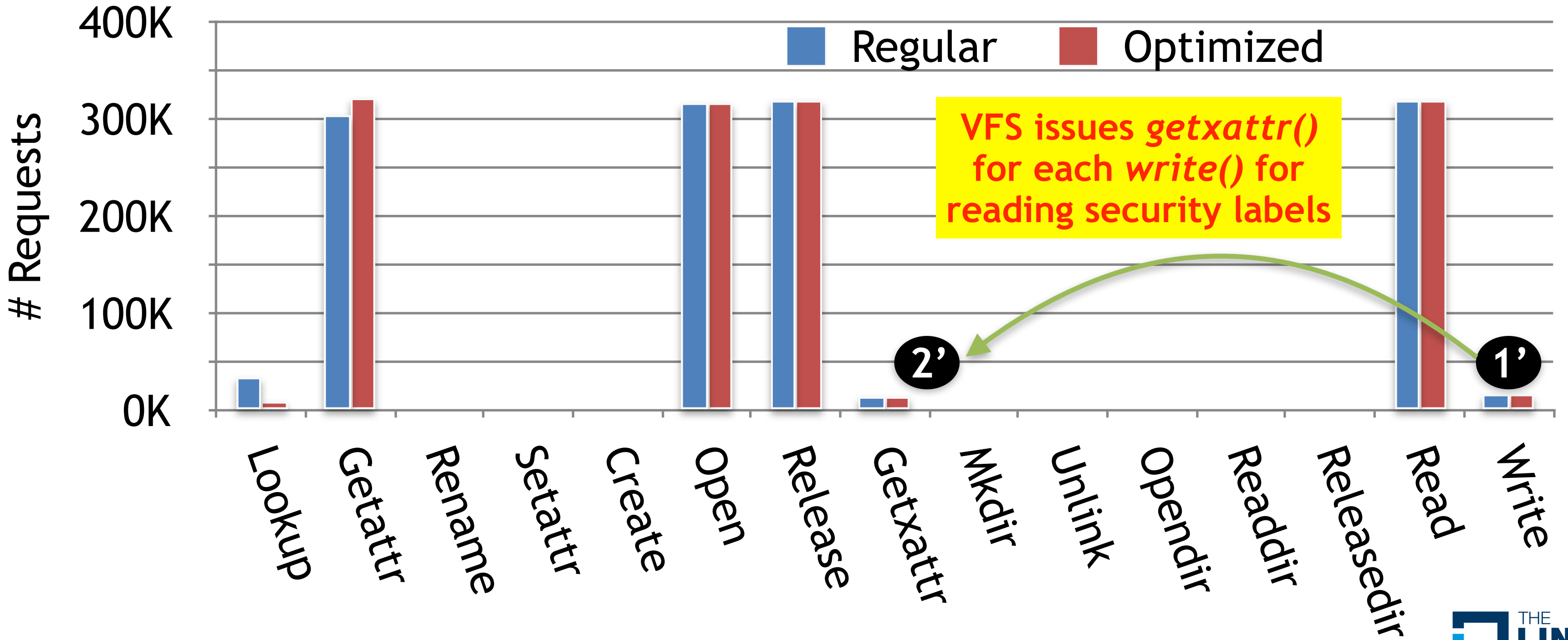
- ***"cd linux-4.17; make tinyconfig; make -j4"***



THE LINUX FOUNDATION

Requests received by FUSE daemon

- ***"cd linux-4.17; make tinyconfig; make -j4"***



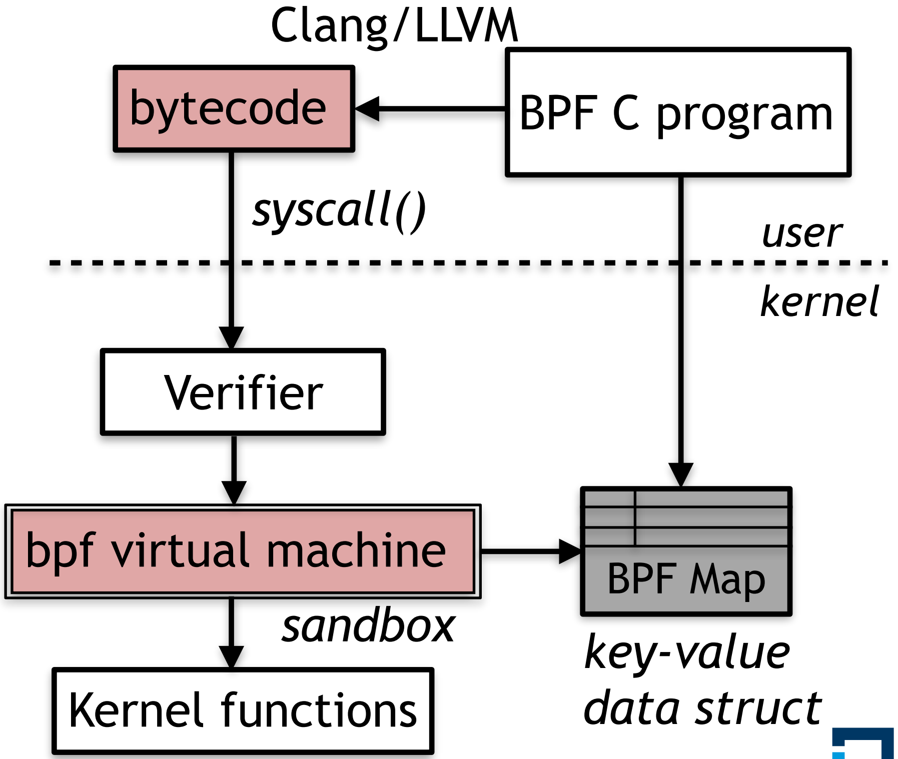
eBPF

- Berkeley Packet Filter (BPF)
  - Pseudo machine architecture for packet filtering

- eBPF enhances BPF
  - Evolved as a generic **kernel extension framework**
  - Used by tracing, perf, and network subsystems

eBPF Overview

- Extensions written in C
- Compiled into BPF code
- Code is verified and loaded into kernel
- Execution under virtual machine runtime
- Shared BPF maps with user space



THE LINUX FOUNDATION

eBPF Simplified Example

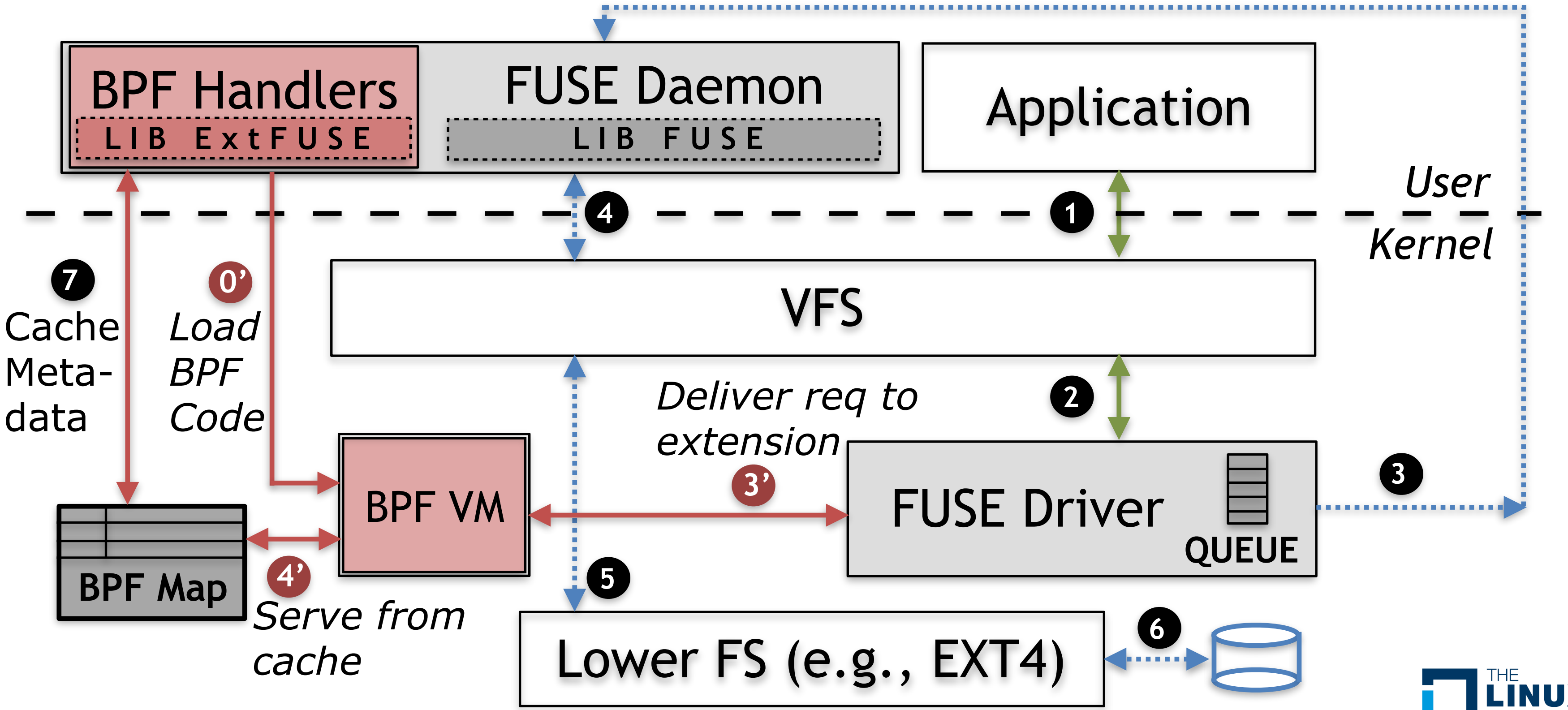
```
struct bpf_map_def map = {
  .type = BPF_MAP_TYPE_ARRAY,
  .key_size = sizeof(u32),
  .value_size = sizeof(u64),
  .max_entries = 1, // single element
};

// tracepoint/syscalls/sys_enter_open
int count_open(struct syscall *args) {
  u32 key = 0;
  u64 *val = bpf_map_lookup_elem(map, &key);
  if (val) __sync_fetch_and_add(val, 1);
}
```

ExtFUSE: eBPF meets FUSE

- Extension framework for File systems in User space
  - Register *“thin”* **extensions** - handle requests in kernel
    - Avoid user space context switch!
  - Share data between FUSE daemon and extensions using BPF maps
    - Cache metadata in the kernel

FUSE Architecture



ExtFUSE Simplified Example

```
struct bpf_map_def map = {
  .type = BPF_MAP_TYPE_HASH,
  .key_size = sizeof(u64), // ino (param 0)
  .value_size = sizeof(struct fuse_attr_out),
  .max_entries = MAX_NUM_ATTRS, // 2 << 16
};

// getattr() kernel extension - cache attrs
int getattr(struct extfuse_args *args) {
  u32 key = bpf_extfuse_read(args, PARAM0);
  u64 *val = bpf_map_lookup_elem(map, &key);
  if (val) bpf_extfuse_write(args, PARAM0, val);
}
```

ExtFUSE Simplified Example

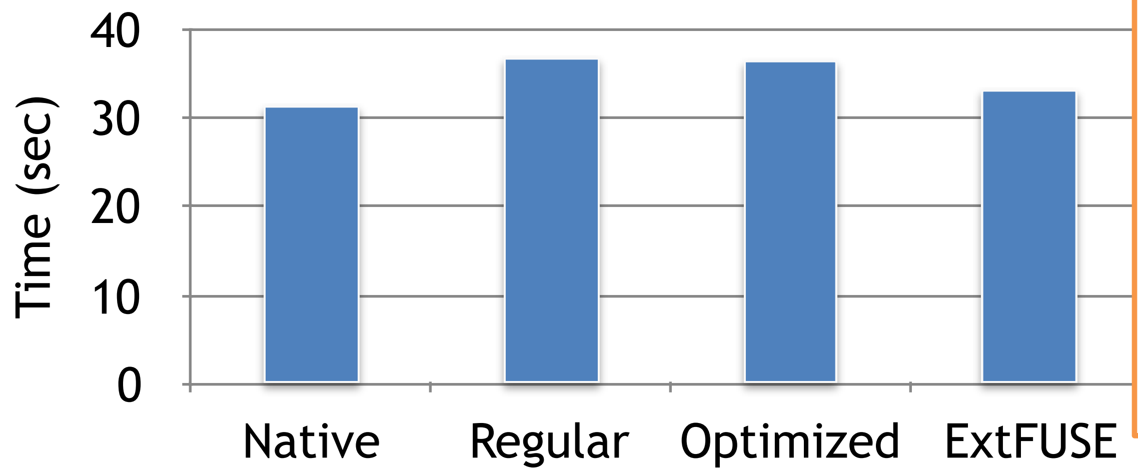
- Invalidate cached attrs from kernel extensions. E.g.,

```
// setattr() kernel extension - invalidate attrs
int setattr(struct extfuse_args *args) {
  u32 key = bpf_extfuse_read(args, PARAM0);
  if (val) bpf_map_delete_elem(map, &key);
}
```

- Cache attrs from FUSE daemon
  - Insert into map on *atime* change
- Similarly, cache *lookup()s* and *xattr()s*

ExtFUSE Performance

- ***"cd linux-4.17; make tinyconfig; make -j4"***
  - Intel i5-3350 quad core, SSD, Ubun
  - Linux 4.11.0, LibFUSE commit # 386



Overhead

Regular Latency: 17.54%

ExtFUSE Latency: 5.71%

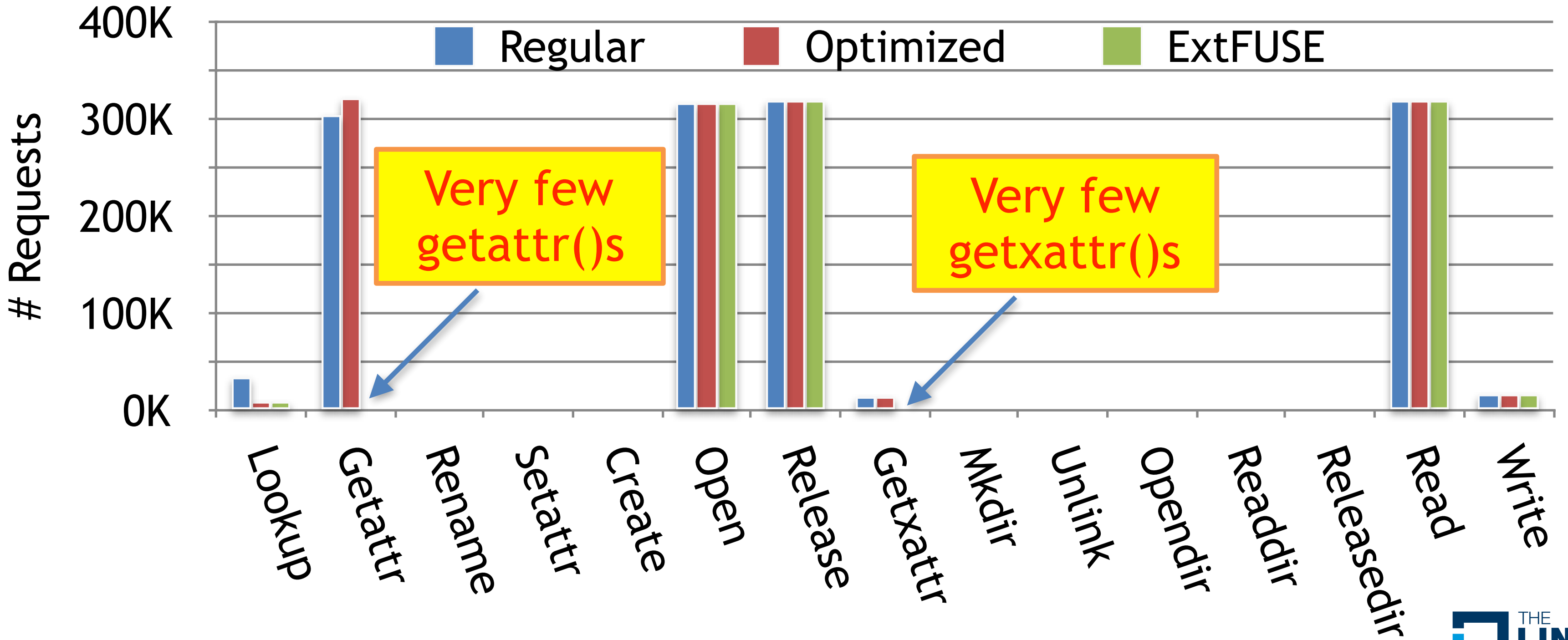
ExtFUSE Memory: 50MB (worst case)

Cached

lookup, attr, xattr

Requests received by FUSE daemon

- ***"cd linux-4.17; make tinyconfig; make -j4"***



THE LINUX FOUNDATION

ExtFUSE Applications

- BPF code to **cache/invalidate meta-data** in kernel
  - Applies potentially to all FUSE file systems
  - e.g., Gluster *readdir ahead* results could be cached

- BPF code to **perform custom filtering or perm checks**
  - e.g., Android SDCardFS *uid* checks in lookup(), open()

- BPF code to **forward I/O requests to lower FS** in kernel
  - e.g., install/remove target file descriptor in BPF map

ExtFUSE Status and References

- Work in progress at Georgia Tech
  - Applying to Gluster, Ceph, EncFS, Android SDCardFS, etc.
  - Project page: https://extfuse.github.io
- References
  - FUSE performance study by FSL, Stony Brook
  - IOVisor eBPF Project
  - BPF Compiler Collection (BCC) Toolchain

THE LINUX FOUNDATION

Thank You!

ashish.bijlani@gatech.edu

www.linkedin.com/in/ashishbijlani

THE LINUX FOUNDATION

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT