ExtFUSE

Extension framework for File systems in User space

Ashish Bijlani, Umakishore Ramachandran
Georgia Institute of Technology

Kernel vs User File Systems

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

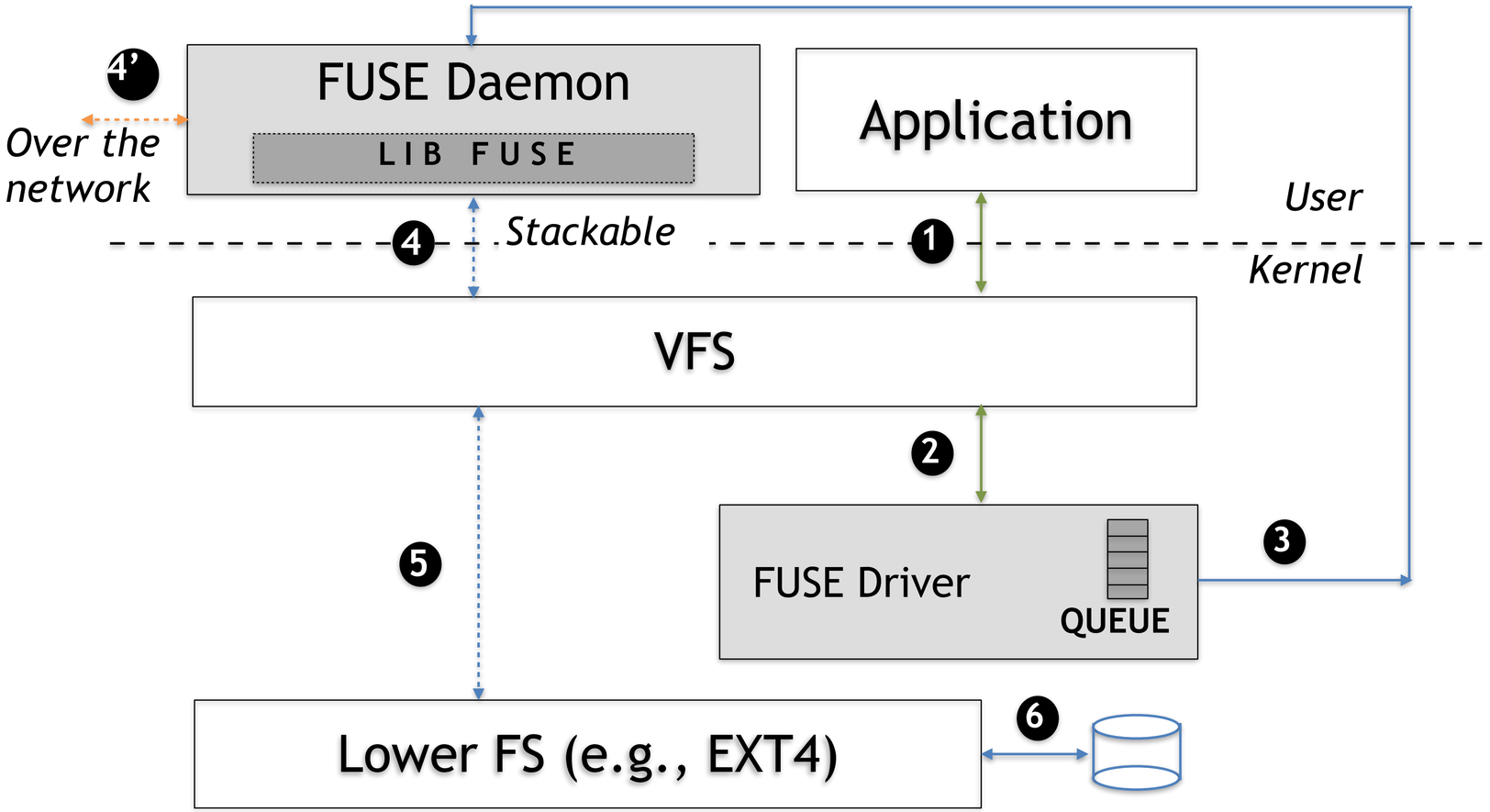
- Examples
  - EncFS, Gluster, etc.
- **Pros**
  - Improved security/reliability
  - Easy to develop/debug/ maintain
- **Cons**
  - **Poor performance!**

File Systems in User Space (FUSE)

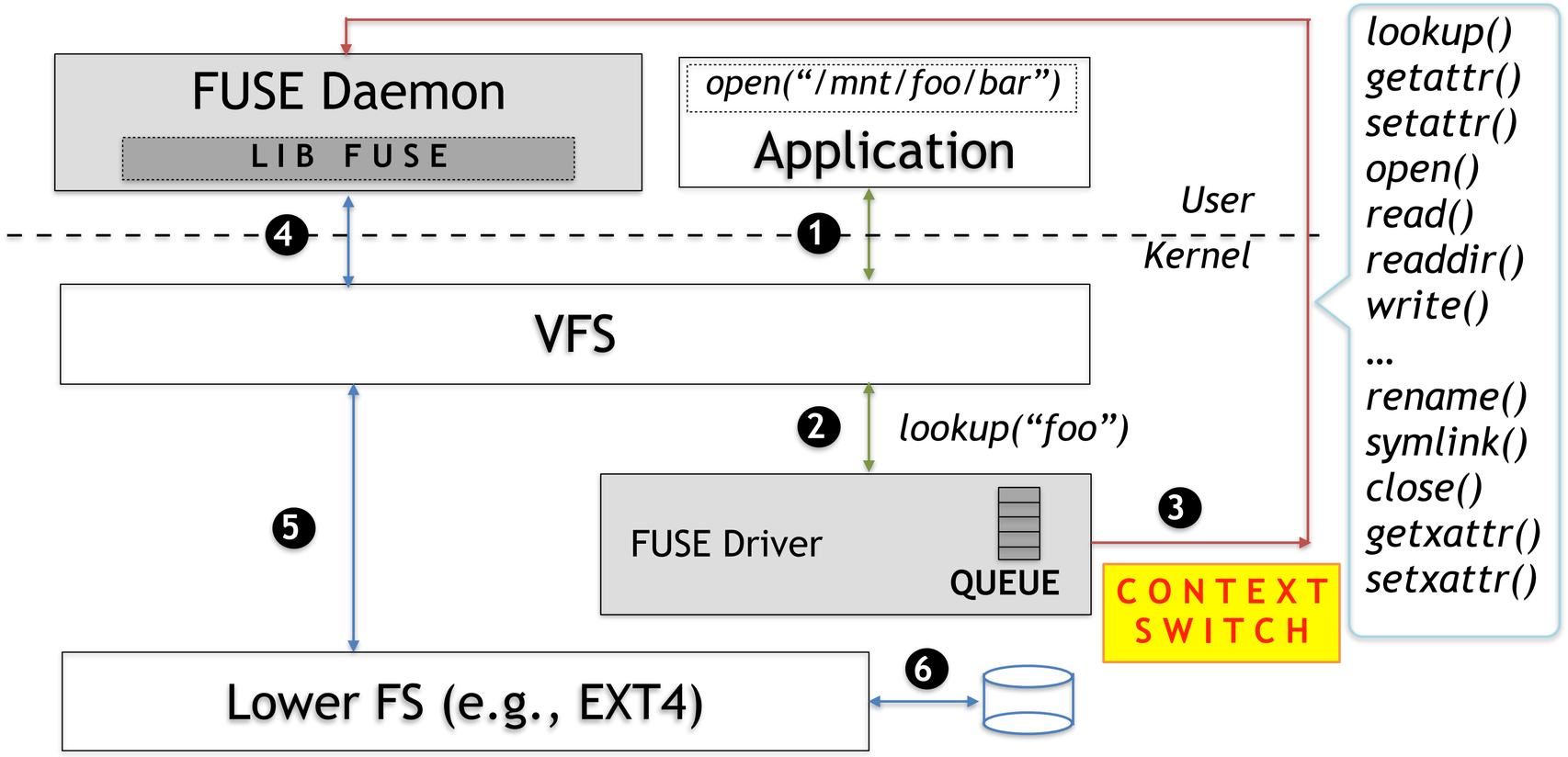
- State-of-the-art framework
  - All file system handlers implemented in user space
- Over 100+ FUSE file systems
  - Stackable: Android SDCardFS, EncFS, etc.
  - Network: GlusterFS, Ceph, Amazon S3FS, etc.

```
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



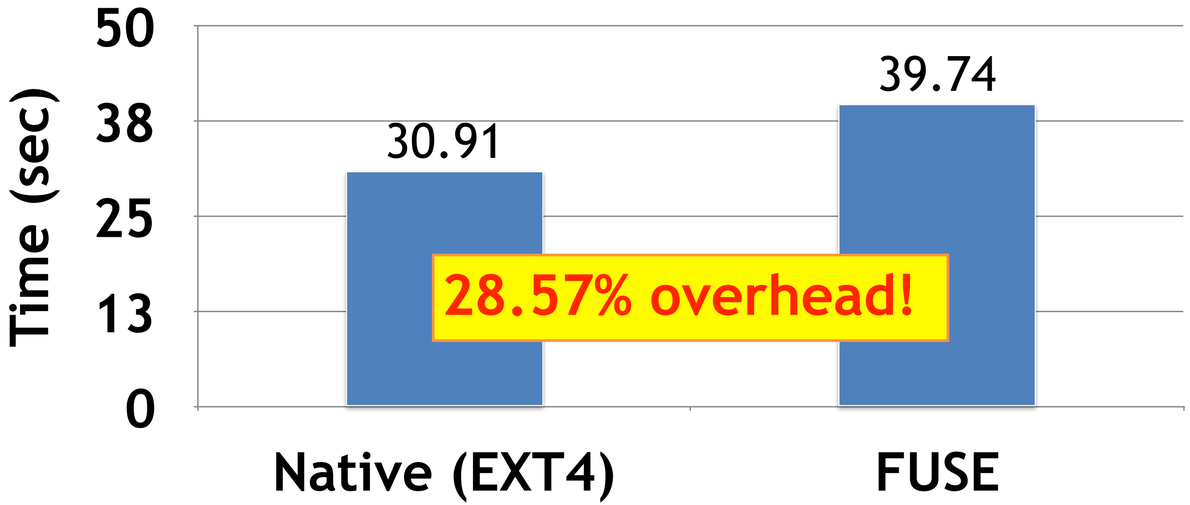
FUSE Architecture



FUSE Performance

"cd linux-4.18; make tinyconfig; make -j4"

- Intel i5-3350 quad core, Ubuntu 16.04.4 LTS
- Linux 4.11.0, LibFUSE commit # 386b1b, StackFS (w/ EXT4)

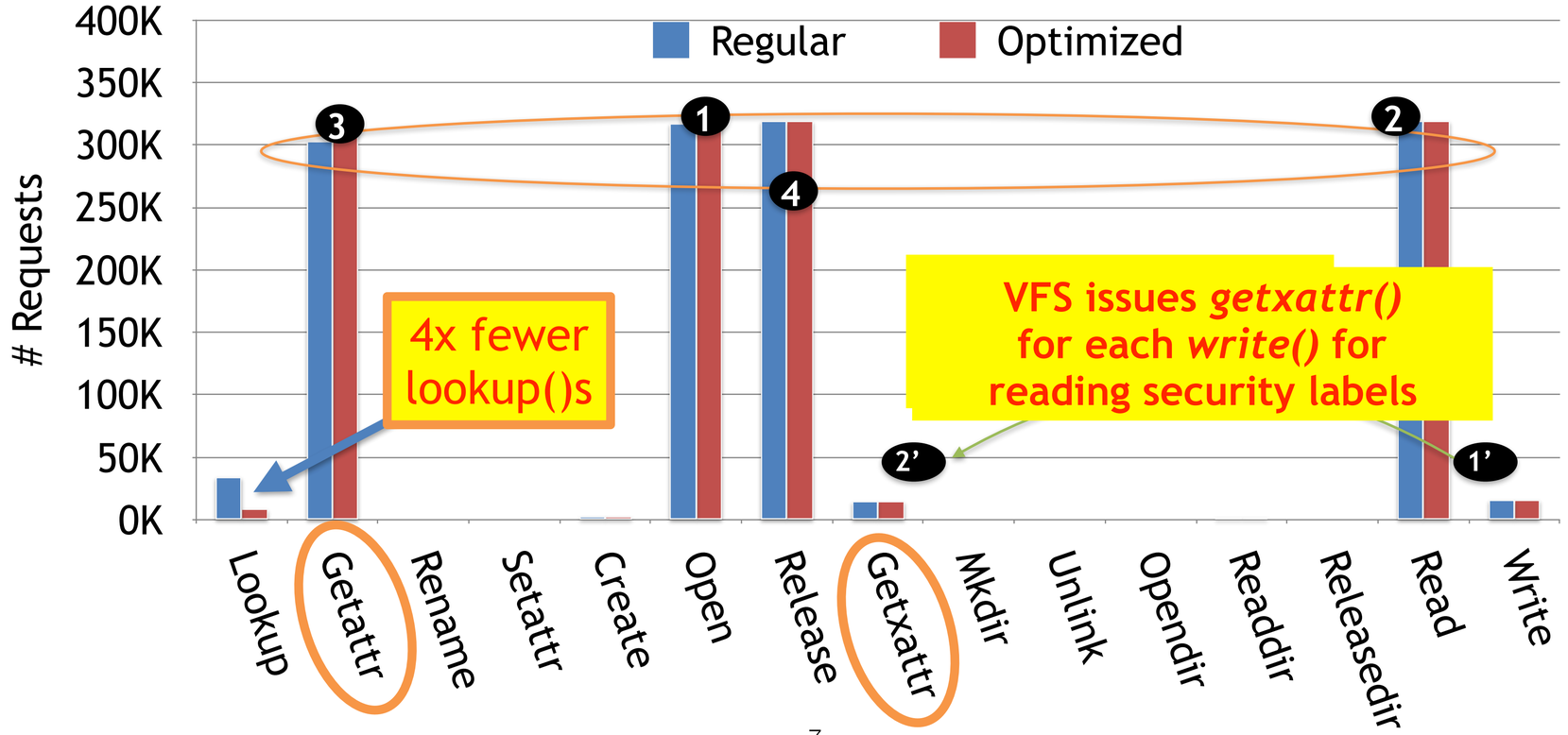


Opts Enabled

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

Req received by FUSE

- ***"cd linux-4.18; make tinyconfig; make -j4"***



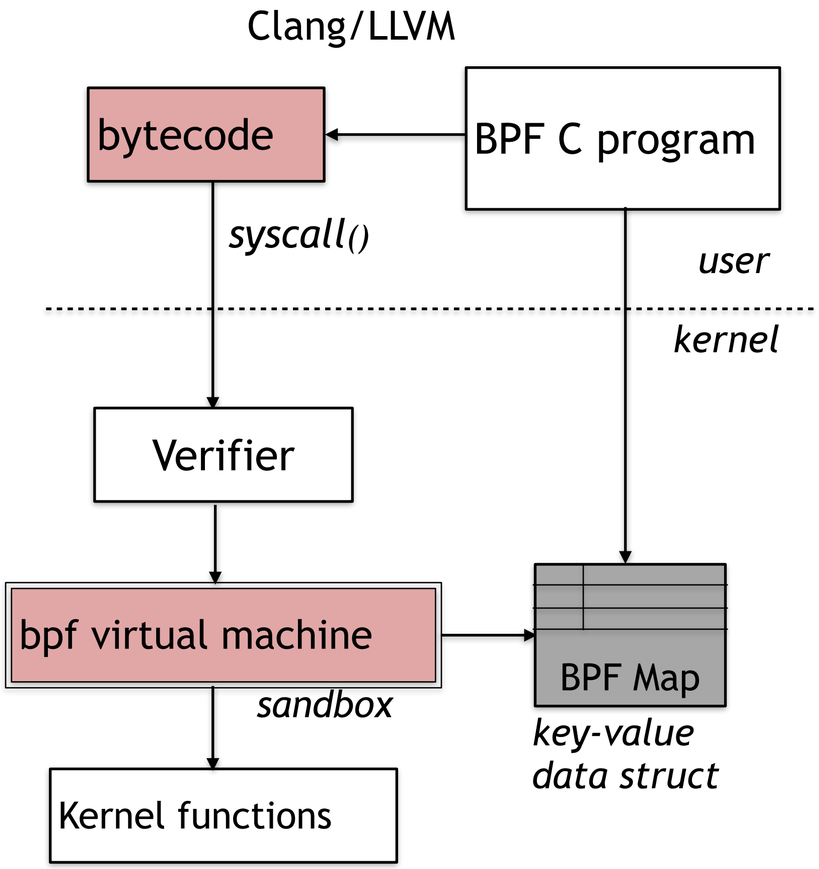
eBPF Overview

- Pseudo machine architecture

- C code compiled into BPF code
- Verified and loaded into kernel
- Executed under VM runtime

- Evolved as a generic **kernel extension framework**
- Used by trace, perf, net subsystems

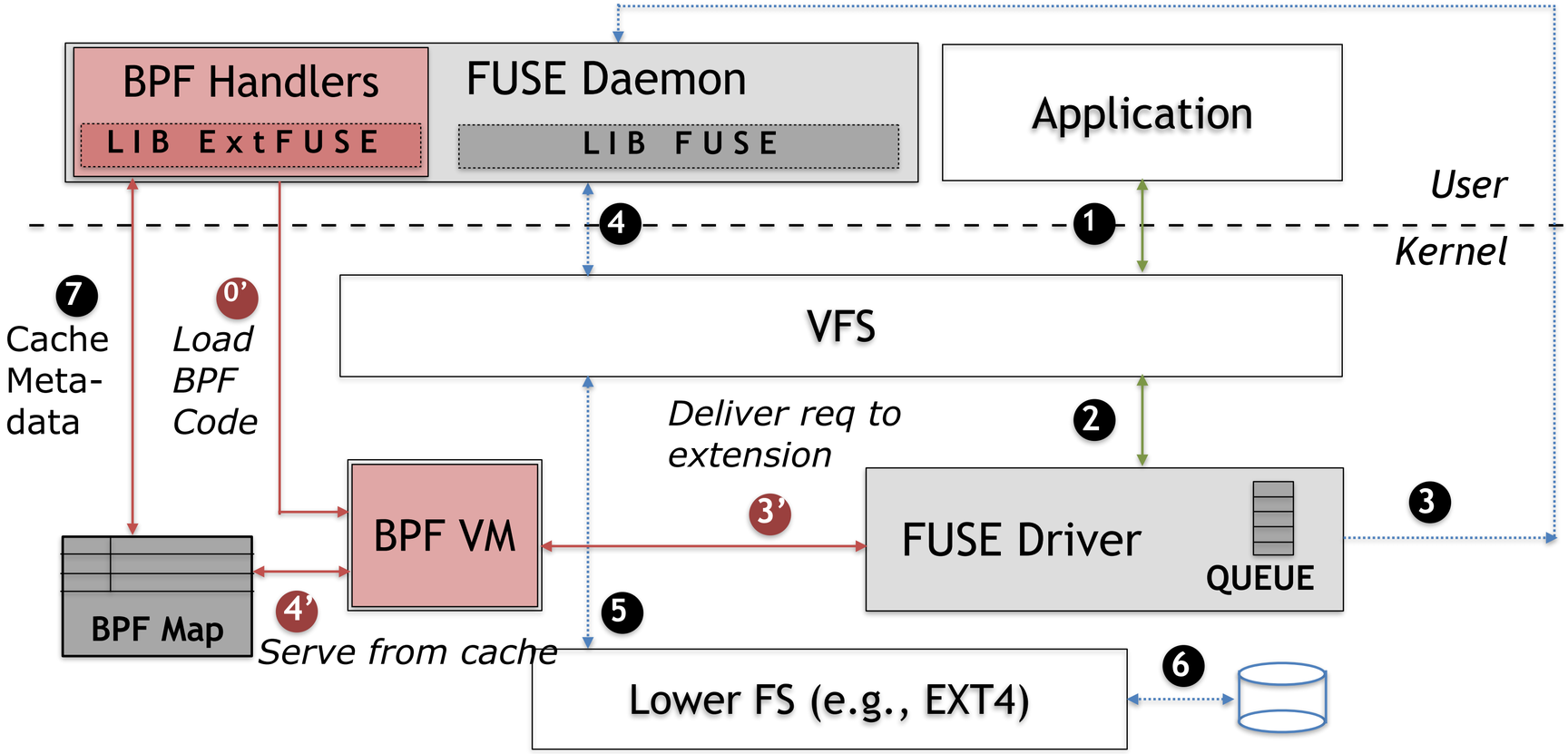
- Shared BPF maps with user space



ExtFUSE

- 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

ExtFUSE Architecture



ExtFUSE Applications

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

ExtFUSE 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 Example

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

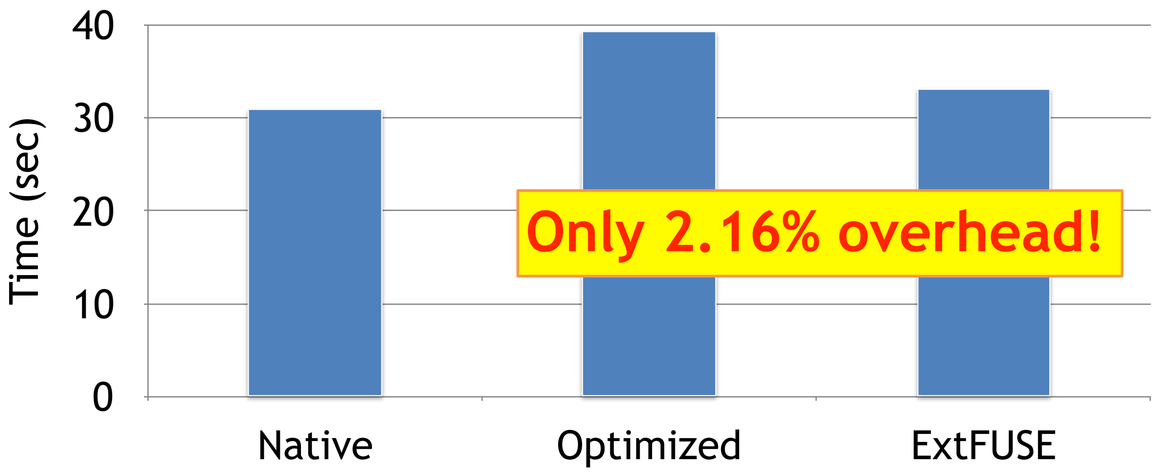
```
// setattr() kernel extension – invalidate attrs
int setattr(struct extfuse_args *args) {
  u32 key = bpf_extfuse_read(args, PARAM0);
  if (val) bpf_map_delete_elem(map, &key);
}
```

- Cache attrs from FUSE daemon
  - Insert into map on *atime* change

- Similarly, cache *lookup()s* and *xattr()s, symlink()s*

ExtFUSE Performance

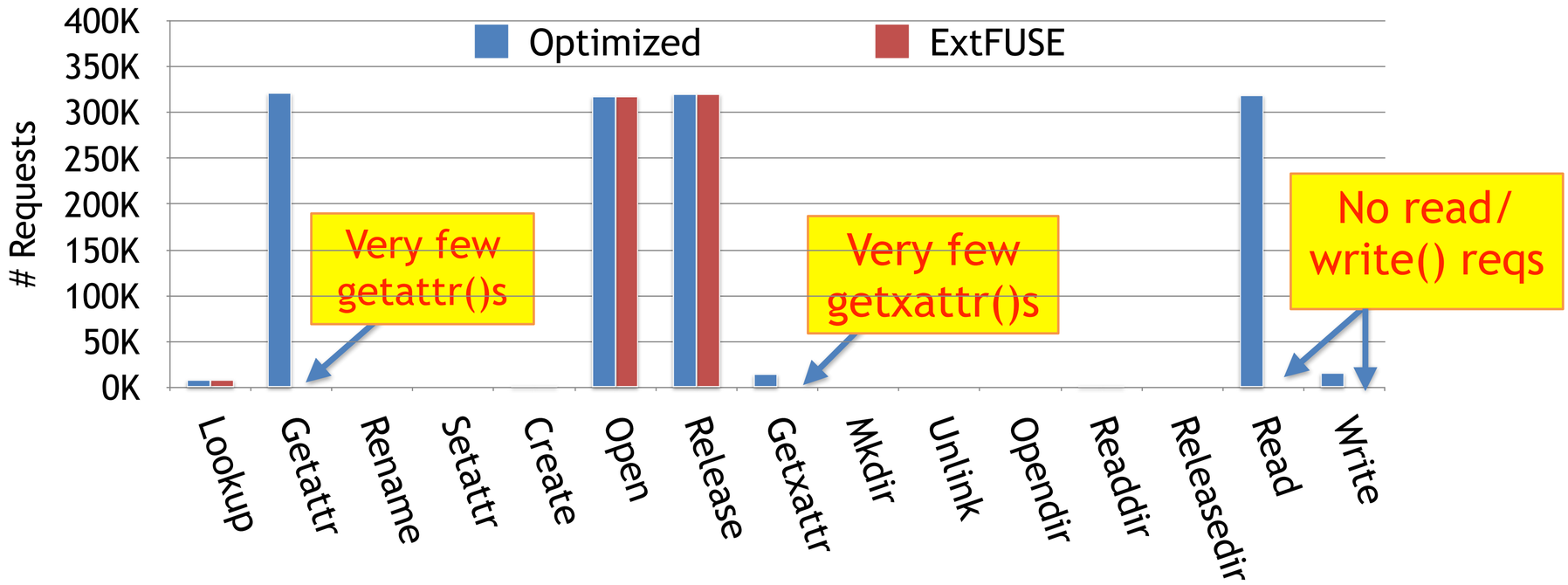
- ***"cd linux-4.18; make tinyconfig; make -j4"***
  - Intel i5-3350 quad core, SSD, Ubuntu 16.04.4 LTS
  - Linux 4.11.0, LibFUSE commit # 386b1b, StackFS (w/ EXT4)



Overhead
Optimized Latency: **28.57%**
ExtFUSE Latency: **2.16%**
ExtFUSE Memory: **50MB**
(worst case)
Cached: lookup, attr, xattr
Passthrough: read, write

Req received by FUSE

- ***"cd linux-4.18; make tinyconfig; make -j4"***



Conclusion

- ExtFUSE framework safely executes “thin” file system handlers in the kernel.
- Developers can use ExtFUSE to
  - Cache metadata requests
  - Directly pass I/O requests to lower FS.
  - Insert custom security checks in the kernel.
- We ported four FUSE file systems to ExtFUSE, including Android SDcardFS and show significant performance improvements.

Thank You!

- Open Source Summit ’18
  - Presentation slides
- Linux Plumbers Conference ’18 talk
  - Presentation video
- Project page
  - https://extfuse.github.io