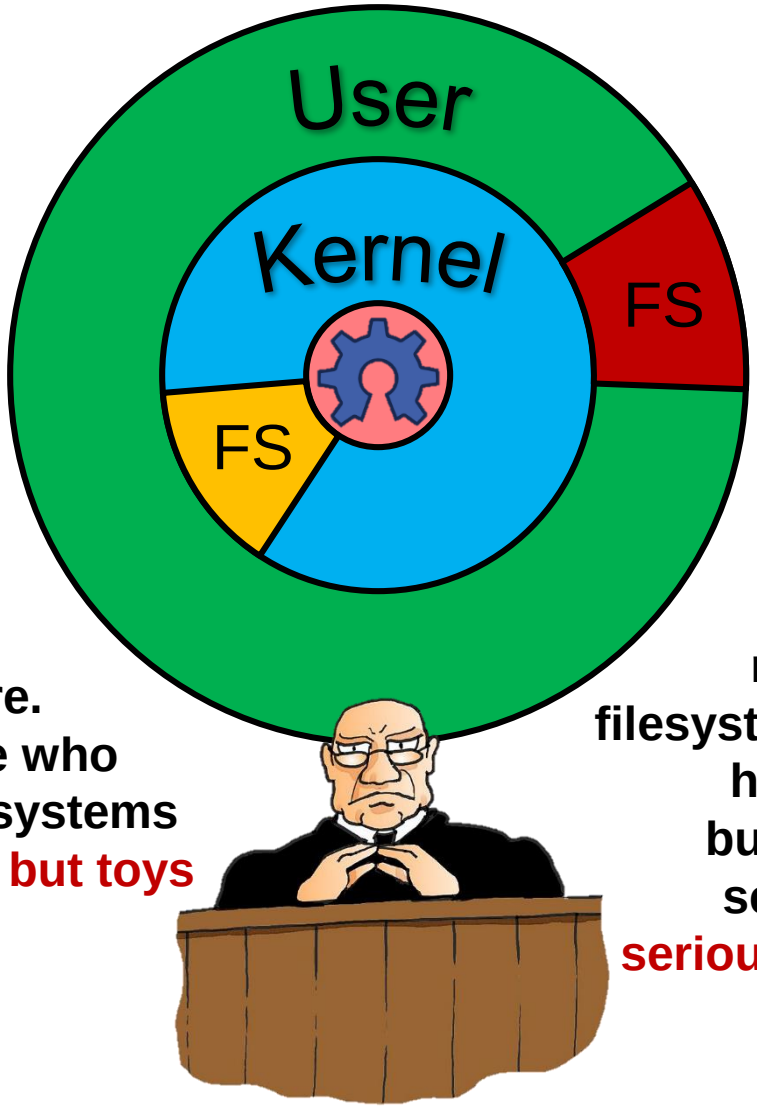
Terra Incognita: On the Practicality of User-Space File Systems

Vasily Tarasov,[✶] Abhishek Gupta,[×] Kumar Sourav,[×] Sagar Trehan[⅄], Erez Zadok[×]

[✶]IBM Research – Almaden , [×]Stony Brook University, [⅄]Nimble Storage

IBM Research nimble storage Stony Brook University

Overview



"Userspace filesystem? The problem is right there. Always has been. People who think that userspace filesystems are realistic for anything but toys are just misguided."

Linus Torvalds

"FUSE has definitely made it easier to write filesystems and a lot of tyros have made toys with it, but it's also possible for serious people to make serious filesystems with it."

Jeff Darcy

IBM Research nimble storage Stony Brook University

Outline

1. Background
2. Pros and cons
3. Evaluation
4. Conclusions and future work

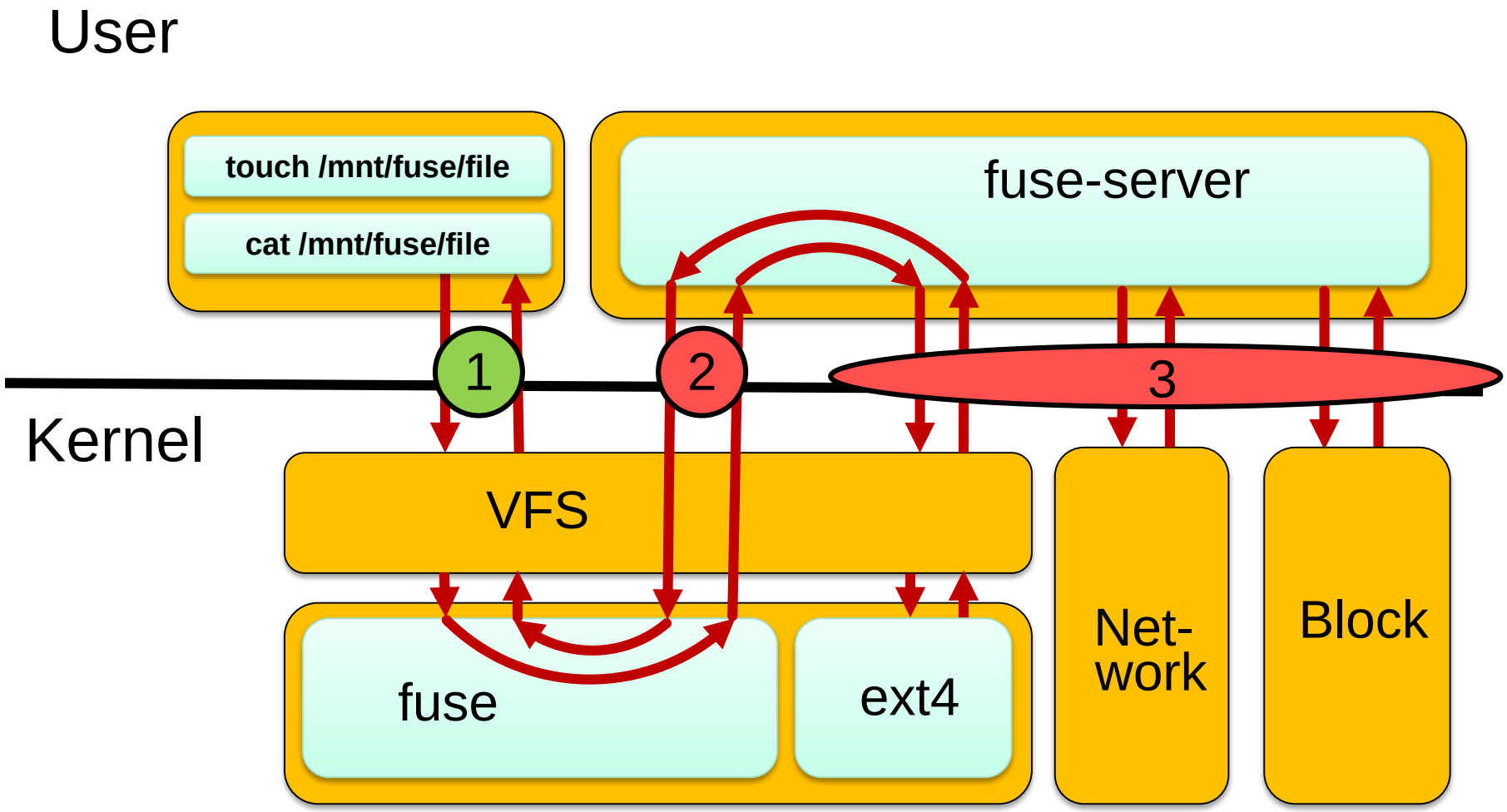
Microkernels?

- Mach3, GNU Hurd, L4
- FUSE is part of the general concept
  - E.g., ufs, ext2, fat, isofs
- Microkernels did not succeed at a time…
- Why FUSE still might succeed?
  - I/O is slow compared to other OS services
  - CPU/RAM are much faster than before
  - New generation of microkernels – L4 [Liedtke90s]

Monolithic Kernels

- Specialized solutions
  - Coda's cache management [Steere90]
  - Arla AFS, IBM GPFS [Westerlund98,Schmuk2002]
- General frameworks
  - Ufo [Alexandrov97]
  - AVFS, Userfs, UserVFS, Podfuk, PerlFS, …
  - **Linux FUSE**
    - 100+ file systems since 2005
    - Kernel: 70 file systems in 23 years

High-level Design



Outline

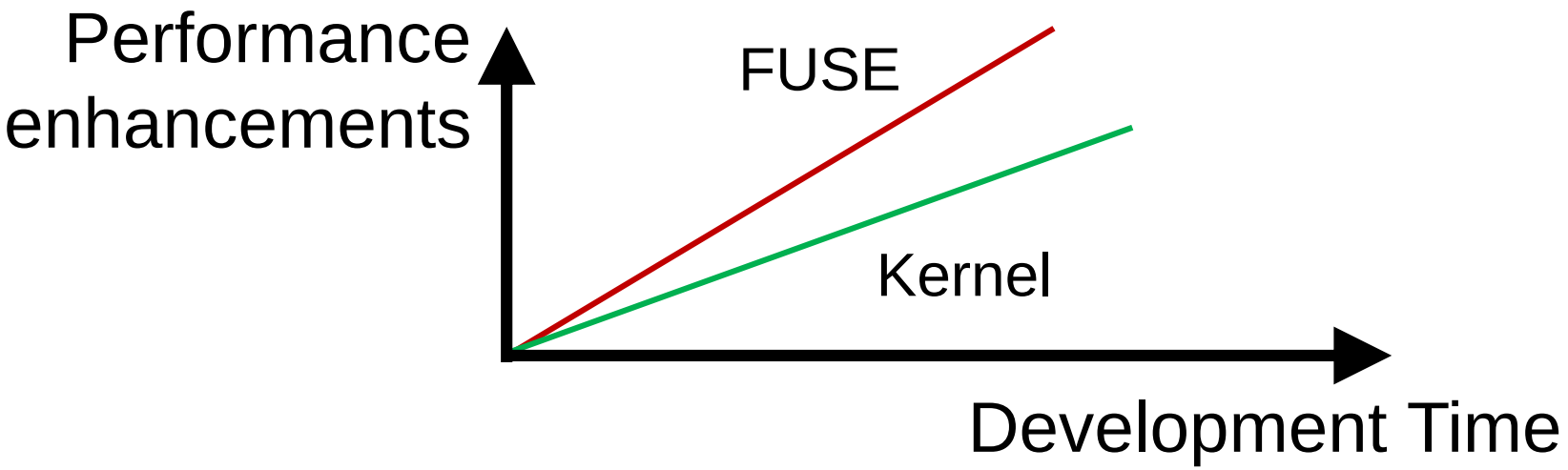
1. Background
2. **Pros and cons**
3. Evaluation
4. Conclusions and future work

FUSE Pros…

- Development ease
  - Toolbox
    - Debug, profile, trace, test
  - Comfy crashes
  - Any programming language
  - A variety of libraries
- Reliability
  - Less code in kernel
- Security
  - User-space is better protected [Kernelis'12]
- Portability

…and Cons?

- Performance
  - Extra memory copying
  - Costly context switches
  - Longer code paths
- …**but** this assumes same performance features!



Outline

1. Background
2. Pros and cons
3. **Evaluation**
4. Conclusions and future work

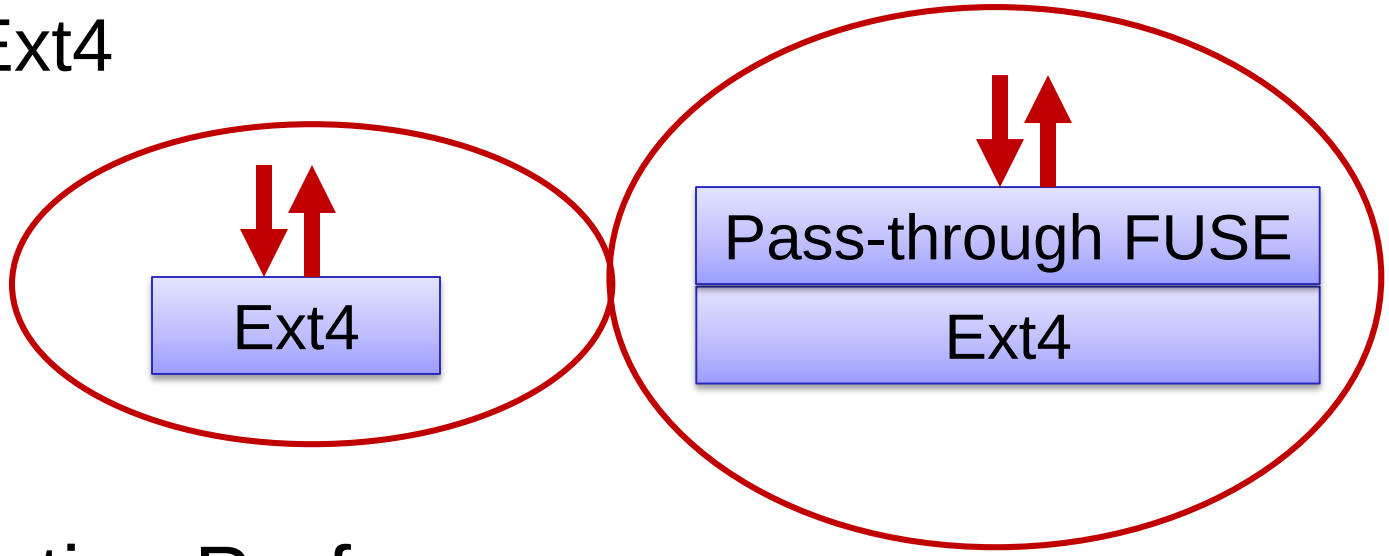
Methodology

- Goal: sample FUSE performance space
- Hardware
  - 15K RPM HDD (4-core 2.4GHz host)
  - Desktop SSD (4-core 2.4GHz host)
  - Enterprise SSD (16-core 3GHz host)
- Workloads
  - Random/sequential read/writes
    - I/O size (4KiB-1MiB), threads (1-32)
  - File creates, deletes
    - Threads (1-32)
  - Web-, Mail-, File-server

} 45 Workloads

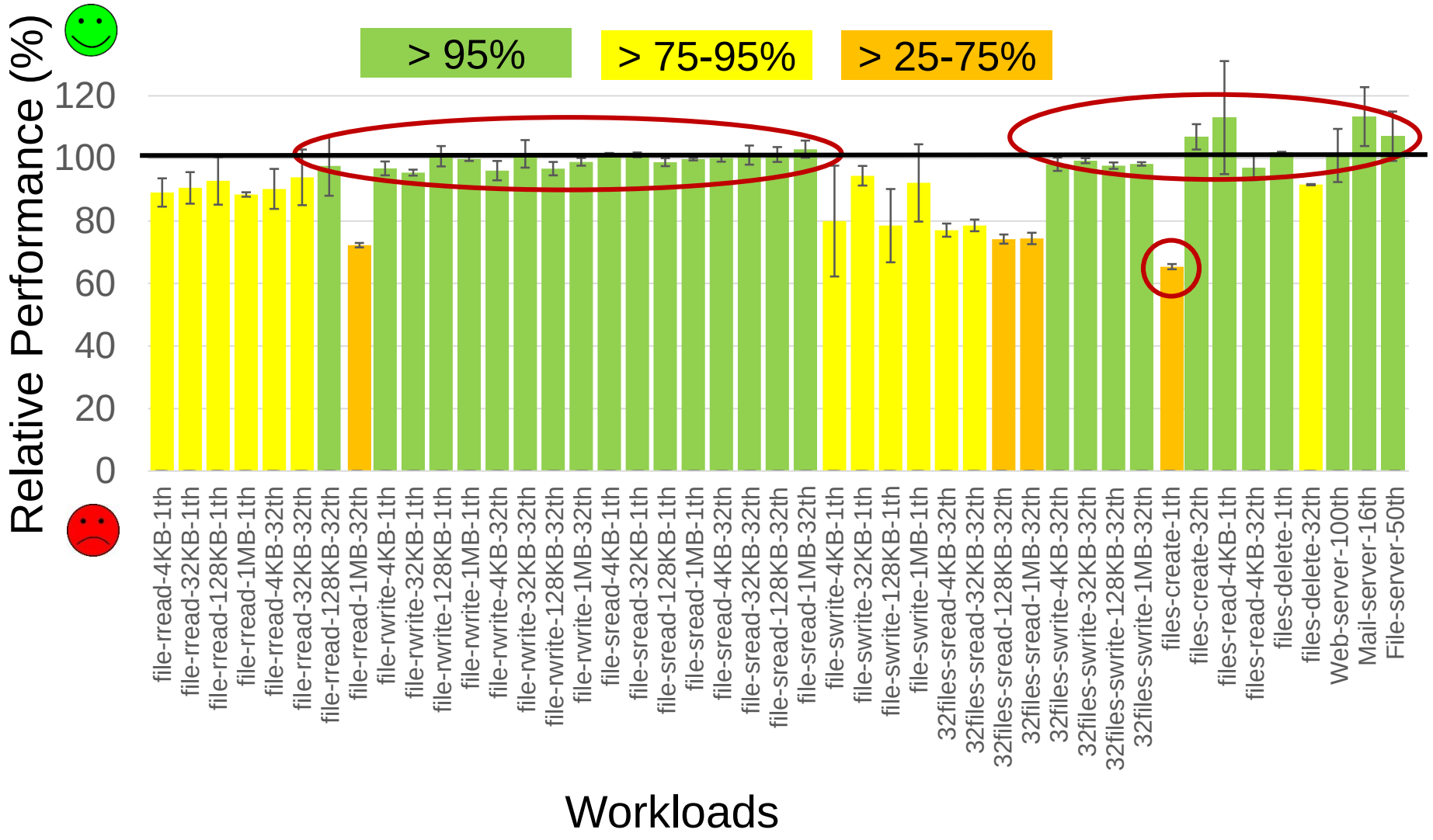
Methodology: Setup

- Software
  - CentOS 7 + Linux 3.19
  - Libfuse-04ad73 (April 2015)
  - Ext4



- Relative Performance
  - Lower boundary

HDD Results



IBM Research nimble storage Stony Brook University

Desktop SSD Results

IBM Research nimble storage Stony Brook University

Outline

1. Background
2. Pros and cons
3. Evaluation
4. **Conclusions and future work**

Conclusions

- FUSE is extensively used
- Controversial opinions on FUSE practicality
- Remains an overlooked research topic
- FUSE is suitable
  - For many workloads
  - For a variety of hardware
- … but not for all!
- Improvements and optimizations are possible

Future Work

- In-depth performance analysis
- In-cache performance
  - Bursty workloads
- Clear boundaries of FUSE applicability
- Optimizations