

An Empirical Study on the Challenges of eBPF Application Development

Mugdha Deokar, Jingyang Men, Lucas Castanheira, Ayush Bhardwaj,
Theophilus A. Benson



ElasticSearch Goes Deep on OpenTelemetry with eBPF Donation

Elastic is collaborating with OpenTelemetry on the common schema and the sensor data model.

Mar 13th, 2024 5:00am by [B. Cameron Gain](#)

Could eBPF Save Us From CrowdStrike-Style Disasters?

In the aftermath of the CrowdStrike outage, enterprises are looking for a safer way to run their IT environments.

Jul 29th, 2024 11:15am by [Steven J. Vaughan-Nichols](#)

eBPF: Meaner Hooks, More WebAssembly and Observability Due

While most enterprises lack the expertise to directly utilize eBPF and should opt for tools configured with eBPF and extended layers of functionality, help is on the way this year.

Feb 9th, 2024 3:00am by [B. Cameron Gain](#)

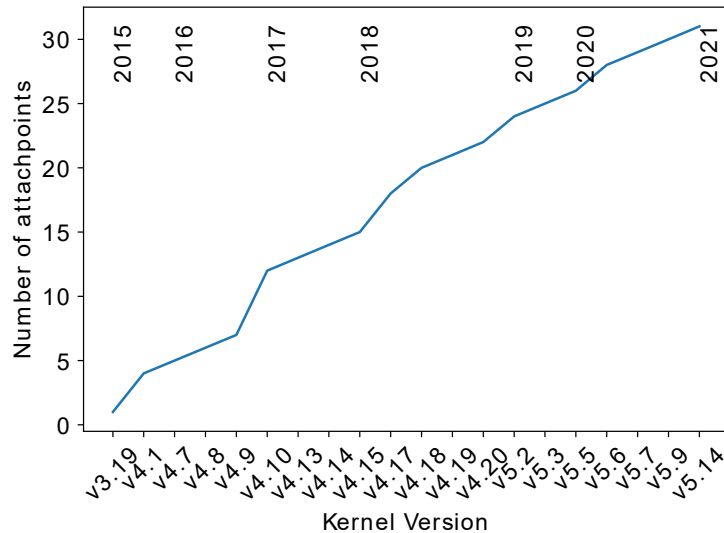


https://thenewstack.com/eBPF-donation/?utm_referrer=https%3A%2F%2Fwww.google.com%2F

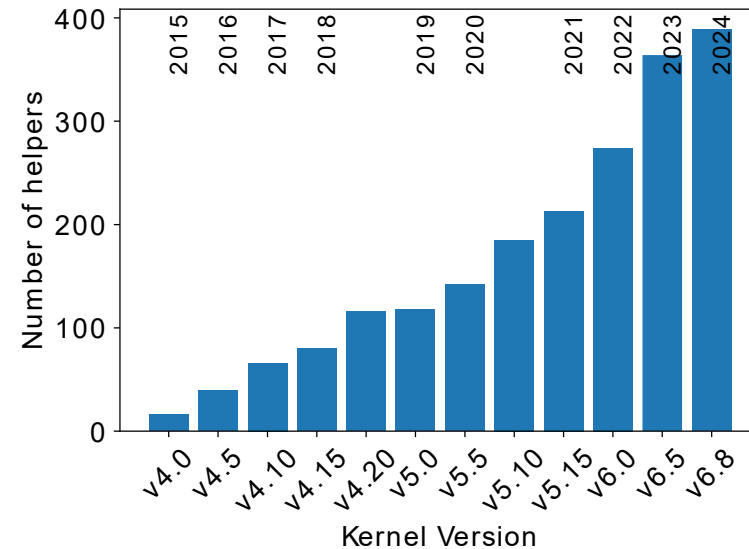


https://thenewstack.com/eBPF-due/?utm_referrer=https%3A%2F%2Fwww.google.com%2F

Ecosystem is Growing More Complex!



100% increase in attach points over the last 6 years



400% increase in helper functions over the last 6 years

This XDP is just 100 lines of code!

A small eBPF/XDP core gives 80% of the value.

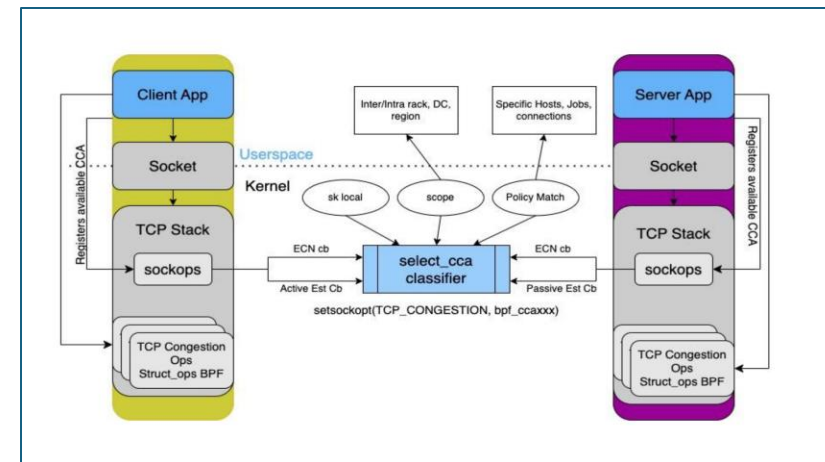
The remaining 80% of the work happens in the user space. It is rarely discussed in talks or articles, as it's thought to be generic enough for all distributed systems. Extra 80% of work is required to support the operations.

Let's delve into the main criteria for employing XDP/eBPF in networking and identify key considerations for your design.



SYSARMOR

- [MITRE](#) ATT&CK framework for coverage
- 50+ bpf hooks
- Rule execution in ebpf code.
- Highly configurable
- Fast execution - optimized for high performance



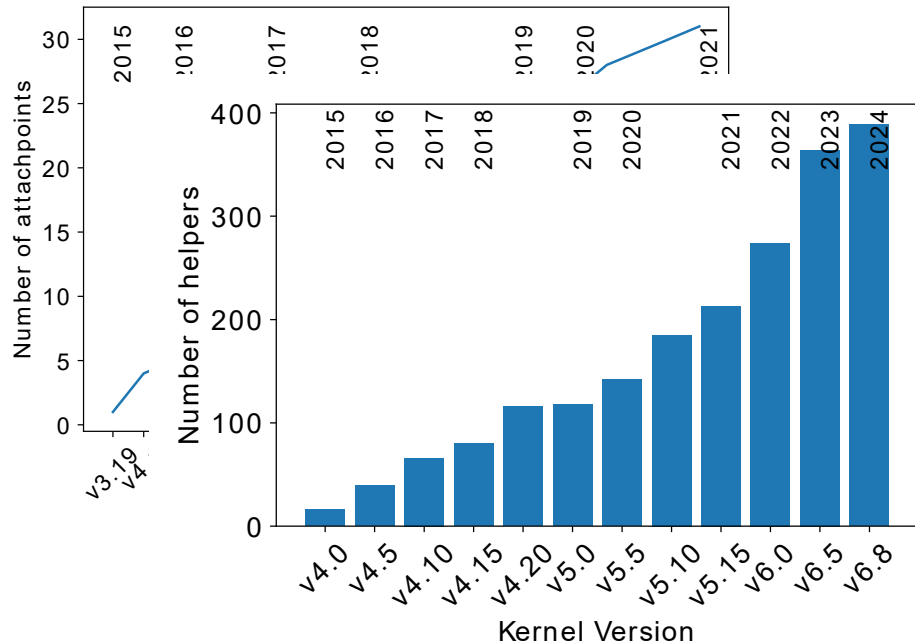
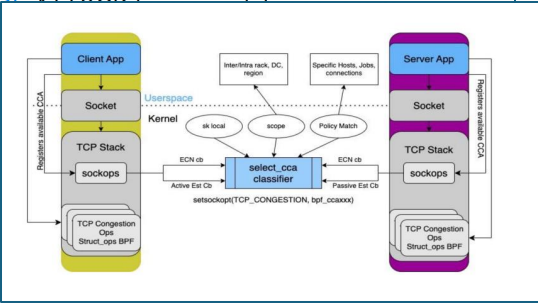


SYSARMOR

WIRE ATTACKS

- 50+

-
- The diagram illustrates the architecture of the `select_oob` classifier, which is designed to handle out-of-band (OOB) traffic. It is divided into two main sections: **Userspace** and **Kernel**.
- Userspace:**
- Client App:** The application on the left side of the diagram. It contains a `Socket` and a `TCP Stack` with `sockops` and `TCP Congestion Ops Struct_ops BPF`.
 - Server App:** The application on the right side of the diagram. It contains a `Socket` and a `TCP Stack` with `sockops` and `TCP Congestion Ops Struct_ops BPF`.
- Kernel:**
- select_oob classifier:** The central component in the kernel. It receives input from the `sockops` of both the Client and Server applications. It is configured with `ECN cb`, `Active Est Cb`, and `Passive Est Cb`. The classifier is associated with the `setsockopt(TCP_CONGESTION, bpf_coaxxx)` system call.
 - Intermediate Data Structures:** The classifier interacts with several data structures:
 - `sk local` and `scope` (ovals) are connected to the classifier.
 - `Policy Match` (oval) is connected to the classifier.
 - `Interfere rack, DC, region` (rectangle) is connected to the classifier.
 - `Specific Hosts, jobs, connections` (rectangle) is connected to the classifier.
- Flow and Data:**
- Registers available CCA:** This data flows from the Client App's `Socket` to the `select_oob classifier` and from the Server App's `Socket` to the `select_oob classifier`.
 - ECN cb:** This data flows from the `select_oob classifier` to the `sockops` of both the Client and Server applications.
 - Active Est Cb:** This data flows from the `select_oob classifier` to the `sockops` of the Client application.
 - Passive Est Cb:** This data flows from the `select_oob classifier` to the `sockops` of the Server application.



XDP programs:

- `crab` - load balancer from the [CRAB proje](#)
- `fw` - firewall from the [hXDP project](#).
- `katran` - load balancer from Facebook. ([s](#)
- `fluvia` - IPFIX Exporter from the [Fluvia pr](#)
- `hercules` - High speed bulk data transfer

Non-XDP eBPF programs:

- `dae` - proxy from the [daeuniverse project](#).

Research Efforts:

- XDP
- KProbes

≡ Example eBPF Programs

- [examples/go-kprobe-counter/](#)
- [examples/go-tc-counter/](#)
- [examples/go-tracepoint-counter/](#)
- [examples/go-uprobe-counter/](#)
- [examples/go-uretprobe-counter/](#)
- [examples/go-target/](#)
- [examples/go-xdp-counter/](#)
- [examples/go-app-counter/](#)

Open source tools (e

- TC
- XDP
- Kprobes
- Tracepoints

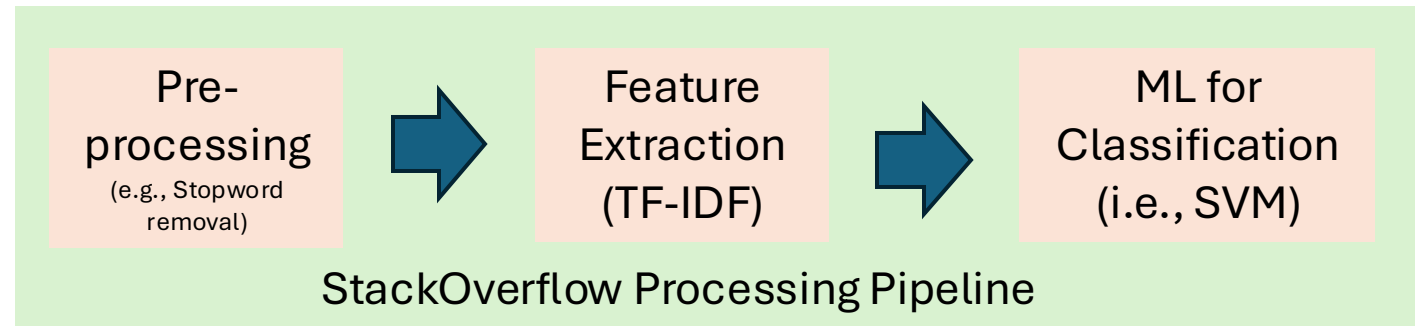
What are the challenges faced in developing and managing eBPF programs?

Software-Engineering Approach to Understanding eBPF Community Challenges



StackOverflow: Problem forum for coding challenges

* 743 Problems tagged as bpf



Classification Dimension

A: Hook point type

B: Ecosystem

C: Development Process

D: Programming Language

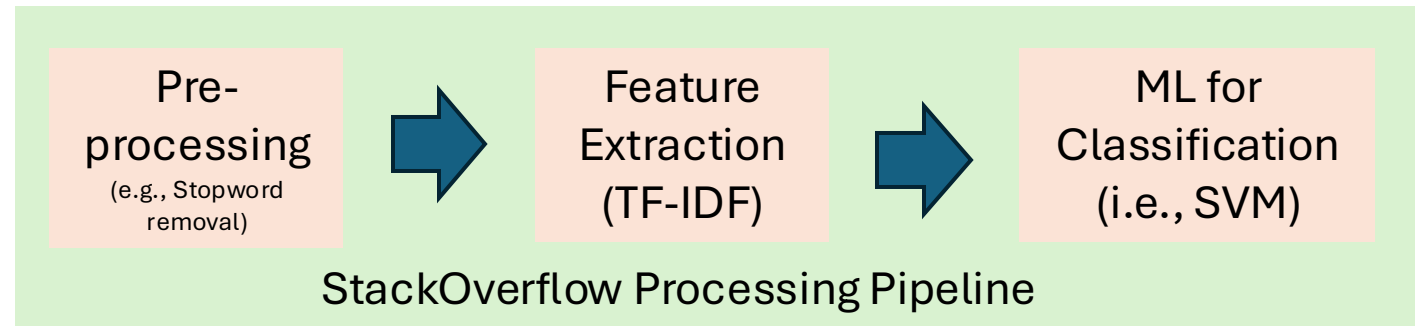
- eBPF Maps
- eBPF Tools and Utilities
- Performance and Optimization
- Kernel Integration and Cgroup Usage
- Error Handling and Verifier Messages

Software-Engineering Approach to Understanding eBPF Community Challenges



StackOverflow: Problem forum for coding challenges

* 743 Problems tagged as bpf



Classification Dimension
A: Hook point type
B: Ecosystem
C: Development Process
D: Programming Language

Model	A	B	C	D
Decision Tree	0.55	0.43	0.90	0.80
SVM	0.68	0.65	0.95	0.87
XGBoost	0.90	0.62	0.90	0.93

Motivating Research Questions



- **RQ1:** How can we lower developer barrier of entry?
- **RQ2:** Which hookpoints is the community grappling with?
- **RQ3:** What is the impact of language choice?
- **RQ4:** How is the Stack Overflow ecosystem addressing eBPF issues?

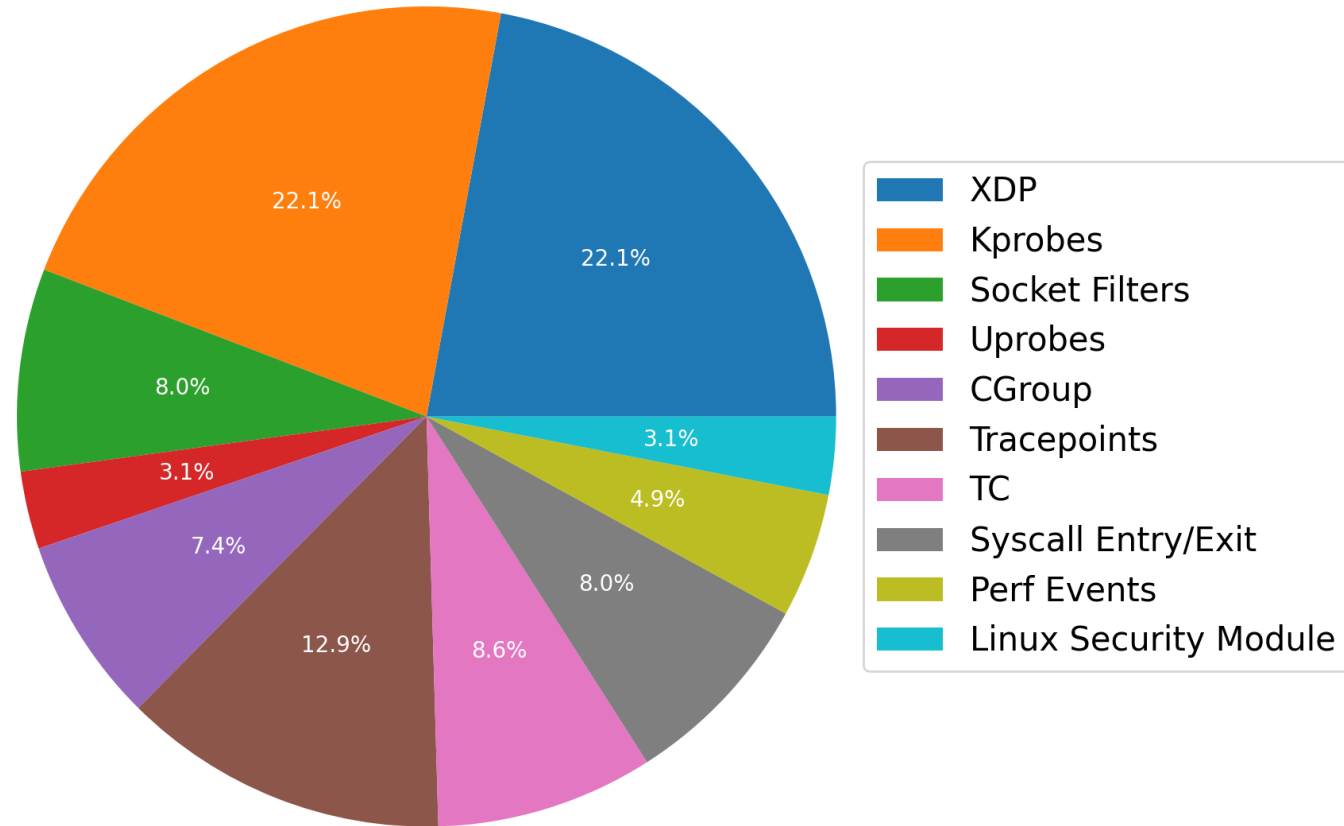
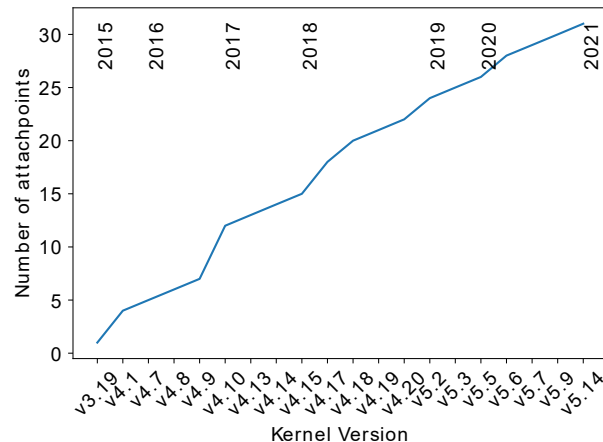
Motivating Research Questions



- **RQ1:** How can we lower developer barrier of entry?
- **RQ2:** Which hookpoints is the community grappling with?
- **RQ3:** What is the impact of language choice?
- **RQ4:** How is the Stack Overflow ecosystem addressing eBPF issues?

Hook Point Usage in the Wild

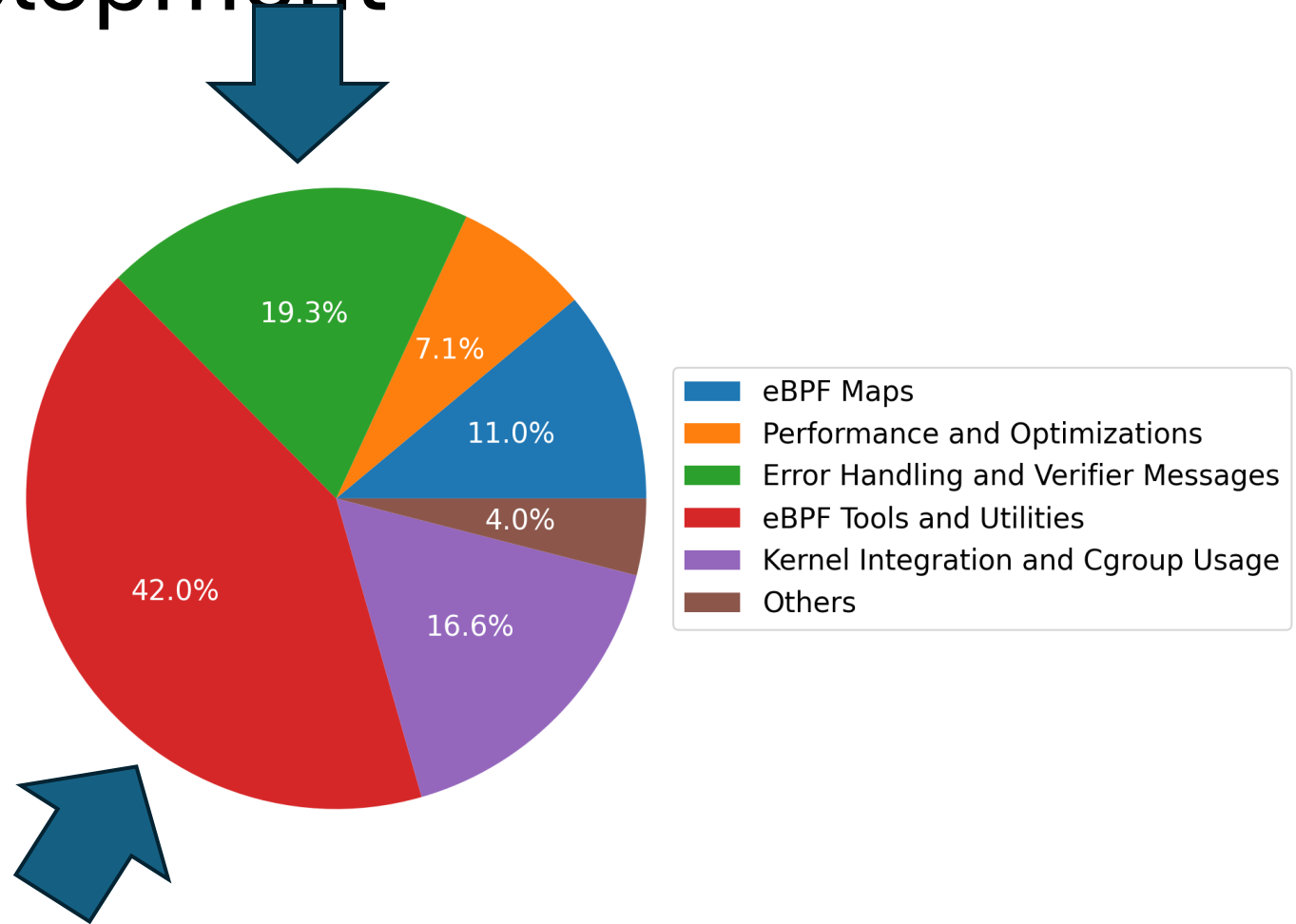
- XDP+Kprobes == 44.2%
 - Non XDP/Kprobe take a lot longer to resolve
- Tracepoint/TC/sockfilters ~ 30%
 - Unsatisfied and unaddressed
 - Research/tools focus on most dominant (older ...)



Barrier to eBPF Development

eBPF Tools/Utilities

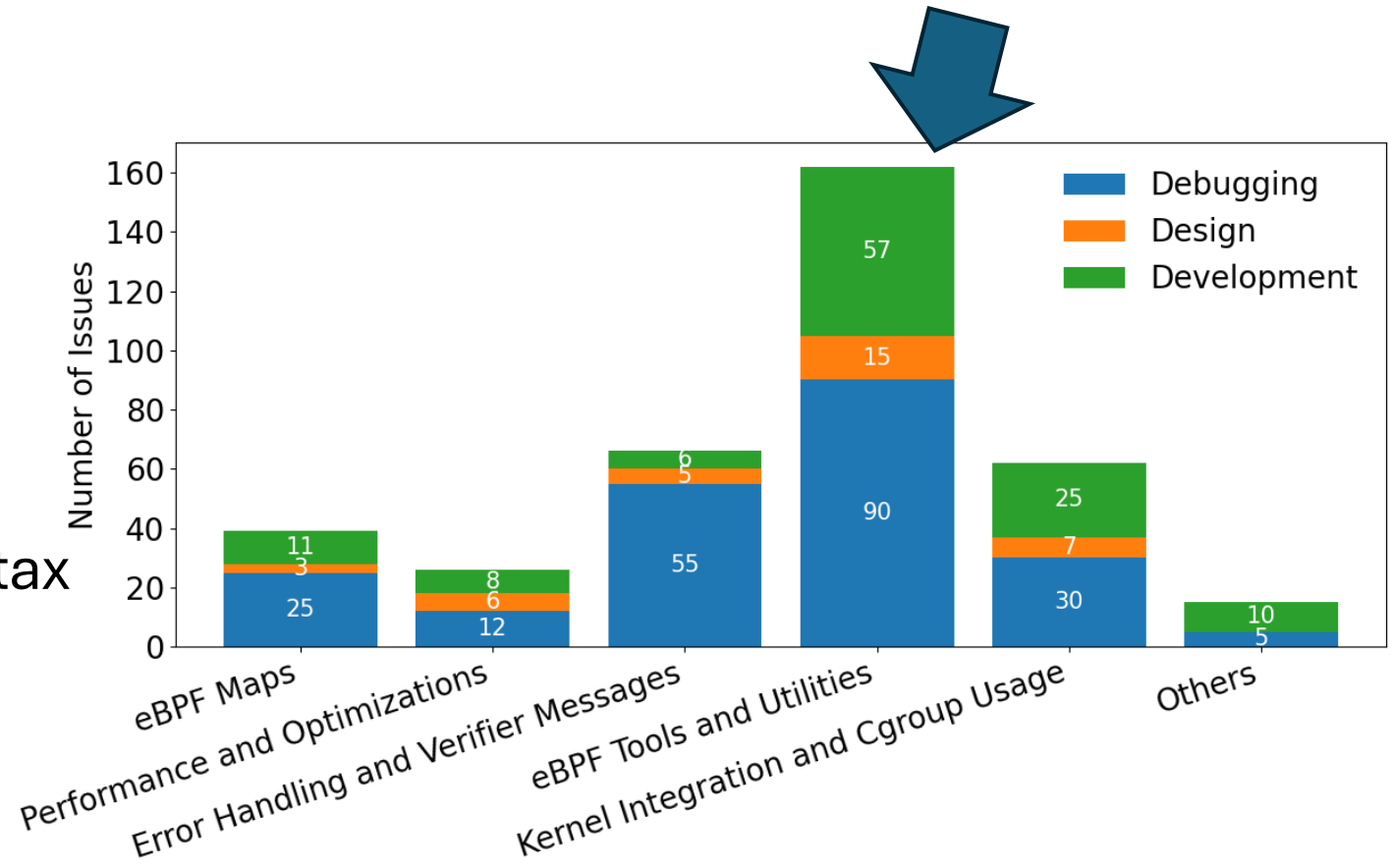
- libbpf/BCC based issues
- Mismatch between documentation and code
- Evolution ==> simplify development but changes syntax
- Easily solved by clarifying mismatch



Hook Point Usage in Context

eBPF Tools/Utilities


- libbpf/BCC based issues
- Mismatch between documentation and code
- Evolution ==> simplify development but changes syntax
- Easily solved by clarifying mismatch



a need for automated patching/updating of documentation (or automated code fixes)

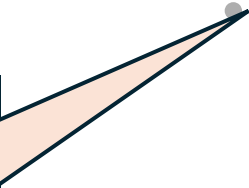
Motivating Research Questions

- **RQ1:** How can we lower developer barrier of entry?

- 
- C still dominates in the kernel.
 - Most other languages are at the user space

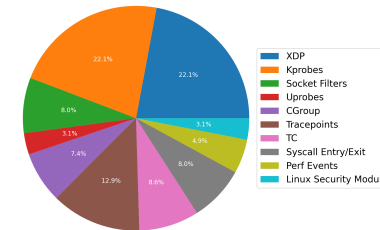
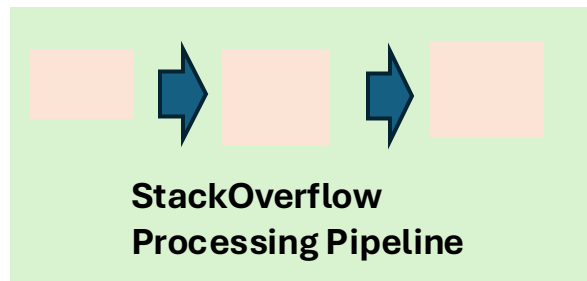
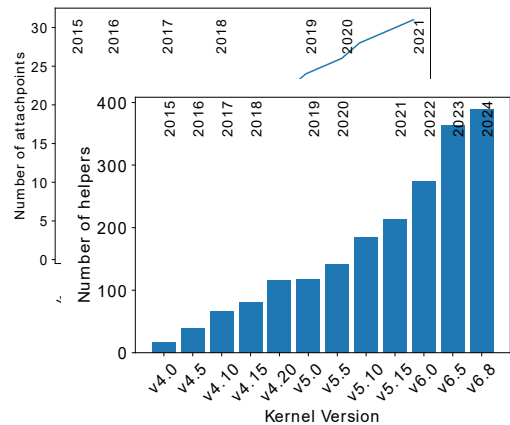
- **RQ2:** Which hookpoints is the community grappling with?

- **RQ3:** What is the impact of language choice?

- 
- Most issues resolved by a handful of users
 - More popular attach points are answer faster

- **RQ4:** How is the Stack Overflow ecosystem addressing eBPF issues?

Conclusion



BPF ecosystem is growing complex

Analyze community challenges

Highlight unmet community needs

Community Survey

theophilus@cmu.edu