The State of eBPF

The 30 million lines of code in the Linux kernel **enables significant functionality,** but adding new functionality can take years.

eBPF has evolved far beyond packet filtering to become a **general purpose computing machine inside the kernel.**

With eBPF, **engineers can quickly build custom programs in the kernel** without the whole community having to accept the change.

In tandem with the needs of cloud native workloads, eBPF **supports capabilities, increases performance, and encourages simplicity.**

eBPF enables **observability** of the Linux system, rewriting or bypassing parts of the **networking stack, and faster vulnerability fixes.**

Big tech companies, including **Google, Meta, and Netflix,** have leveraged eBPF in their data centers for years.

Many applications already use eBPF enabling continuous profiling, monitoring servers, observability platforms, and performance monitoring tools.

Innovation is at the heart of eBPF, creating an iteration cycle that is faster, safer, and allows greater flexibility.

A verifier and JIT compiler provide **safety and performance benefits** in eBPF deployments.

Other challenges to eBPF include the **performance-features tradeoff, co-existence and interoperability of tools,** and the kernel expertise needed to write programs.

The eBPF Foundation and steering committee provide **technical direction and optimize collaboration** on the technology's roadmap.

Standardization is in the future of the eBPF ecosystem, around instruction sets for all operating environments because it is becoming a critical layer in the cloud native infrastructure stack.