

IO Visor: Revolutionizing Kernel I/O & Networking

A non-stop cycle of increasing virtualization, cloud based applications, containers, and microservices are fueling the rapid transformation of data centers. “Densification” of data centers is compounding the need for dynamic, agile infrastructure that is able to scale-out rapidly while providing virtual networking and IO functions seamlessly.

Scheduled down time, manual workflows, error prone operational procedures are cumbersome in the age of DevOps and global competitiveness. While tremendous innovation has swept across infrastructure such as SDN and NFV technologies, challenges remain in being able to achieve the flexibility and performance desired, without compromise.

To address this challenge, IO Visor Project, part of the Linux Foundation, was created to enable developers to build, innovate, and share an open, programmable data plane with dynamic IO and networking functions.

What are the Challenges?

As new initiatives and companies form to address upper layer functions such as creating virtualized network functions (VNFs) or developing multi-protocol software defined controllers, the fundamental design of IO and networking subsystems in Linux hasn’t garnered enough attention. Functions developed to run in user space are not able to achieve high performance and often require hair-pinning traffic flow to perform multiple functions resulting in further performance bottlenecks. Running in a distributed, scale-out environment for web based applications is been a challenge and presents performance problems and high availability concerns.

For developers, creating IO modules in-kernel is a challenge due to lack of virtualization in

the kernel. Loading and unloading IO modules requires modifying, recompiling and even rebooting software. The lack of flexibility and performance is an increasing problem that the industry hasn’t been able to solve, until Extended Berkeley Packet Filter (eBPF) came along.

What is eBPF?

eBPF provides an embedded universal in-kernel virtual machine, extending its use with networking and non-networking functions alike. Through this virtual machine structure, eBPF enables infrastructure developers to create any in-kernel IO module and load/unload them at run time, without recompiling or rebooting.

IO Visor Project

IO Visor Project’s architecture consists of single or multiple IO Visor Engines built on eBPF technology and development tools. The IO Visor Engine has a set of plugins that provide functionality in different areas such as networking, tracing, security, and others.

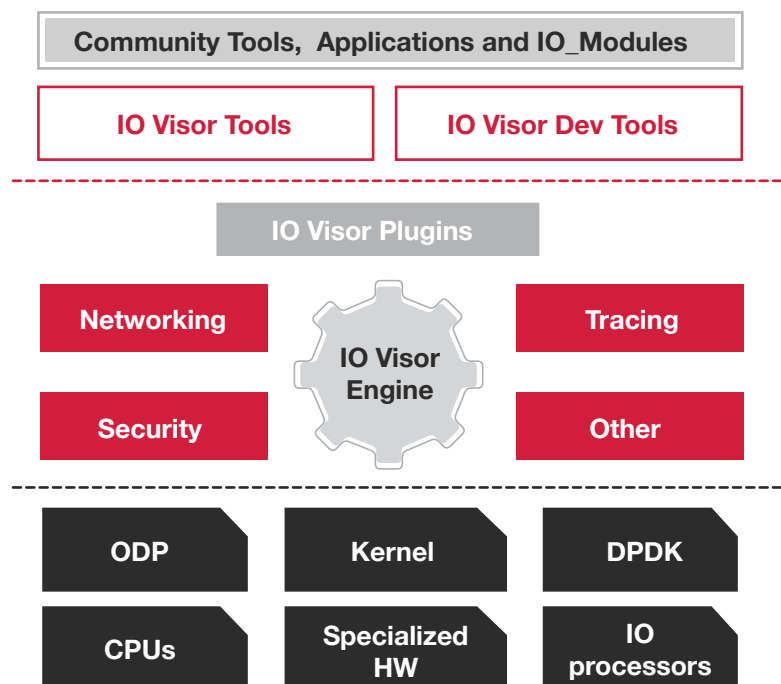


Figure 1: IO Visor

The two key benefits that the IO Visor Project delivers are flexibility and performance.

Flexibility:

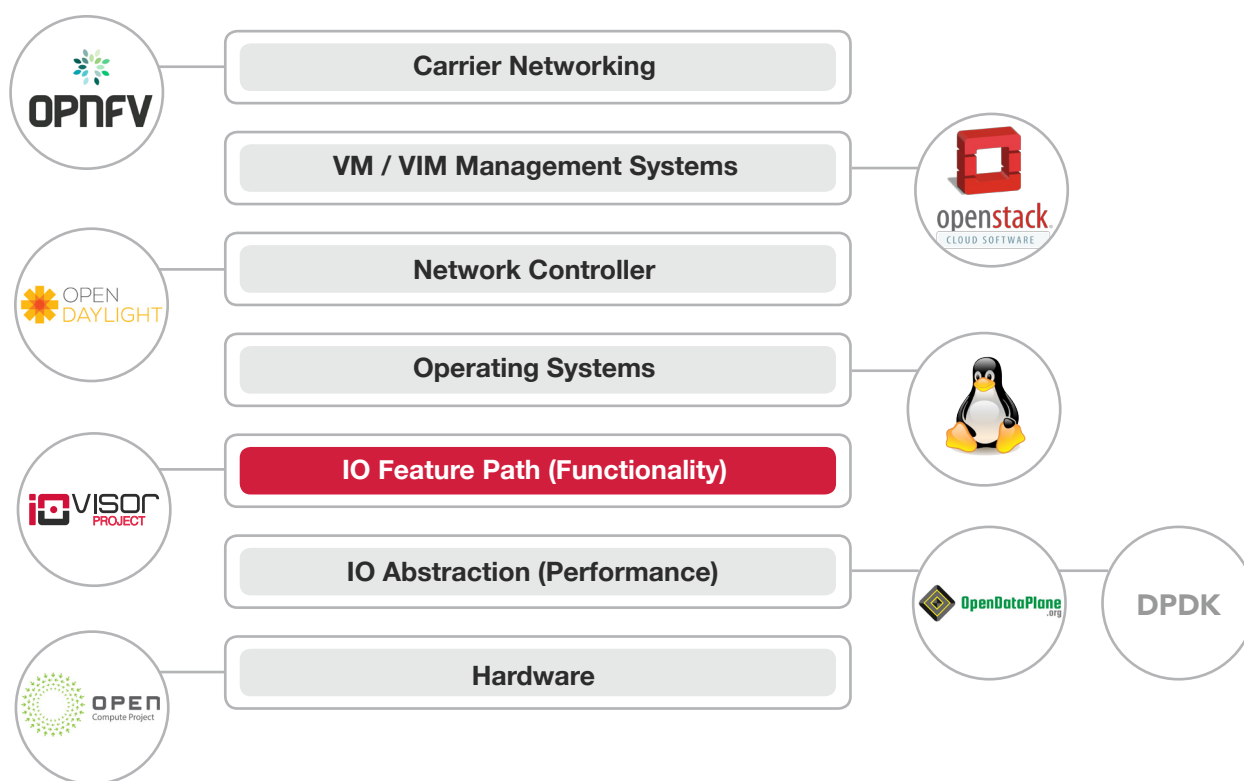
Having a programmable, extensible architecture enables developers to build dynamic IO modules that run in-kernel, loaded and unloaded without any disruption to the system. Furthermore, the IO modules are platform and hypervisor independent, meaning that they could run on any hardware or x86 chipset with Linux.

Performance:

Running IO and networking functions in-kernel delivers the performance of hardware without layers of software. Since the kernel runs in distributed scale-out mode, IO Visor functions in the same way, eliminating hairpinning, tromboning and bottlenecks that are prevalent in so many implementations today.

IO Visor Ecosystem

As a community, the IO Visor Project is supported by companies from a diverse ecosystem across semiconductor, software, and systems. The support from this broad ecosystem is a validation of IO Visor Project's significance in the Linux community and synergy with other upstream and downstream projects. The IO Visor Project works with upstream projects such as OpenDayLight, OpenStack, and OPNFV. Open source initiatives such as ODP, DPDK, and OCP may also leverage IO Visor Project to achieve performance and feature functionality.



Join IO Visor Project

There is an on-going community of developers who have been working together for several years on IO Visor Project. The community leverages Github for developer resources at <https://github.com/iovisor>. It is open to all developers and there is no fee to join or participate. We welcome users, developers, and anyone interested to join at iovisor.org.