Automation, Programmability, Containerization & Integration

The future of networking

Suresh Krishnan, CTO

kaloom™

Agenda

- Introduction
- Automation
- Programmability
- Containerization
- Integration
- Summary

Who am I?

20+ years in the Networking Industry

35+ Standards

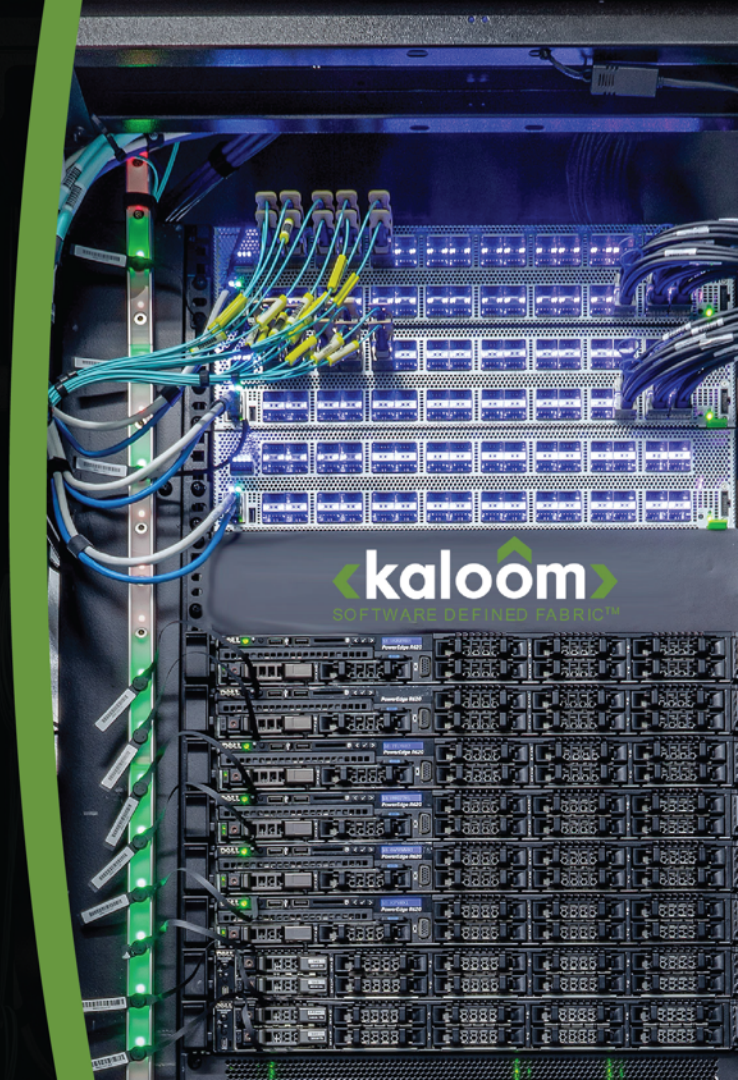
40+ granted Patents

Area director at the IETF

kaloom

Automation

kaloom™

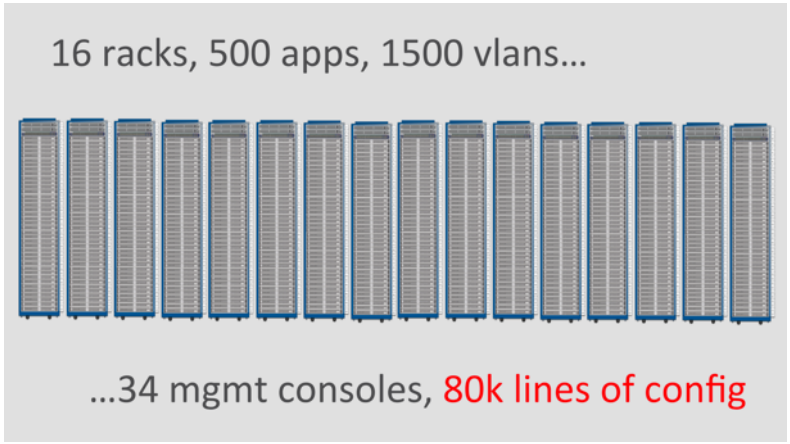


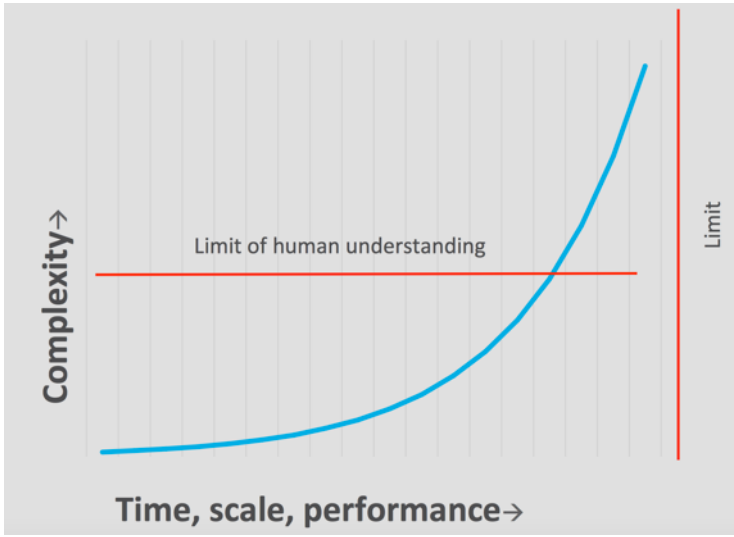
"I love configuring my datacenter networking"

- said no one ever


vivien@lafibre.info

Complexity and human understanding





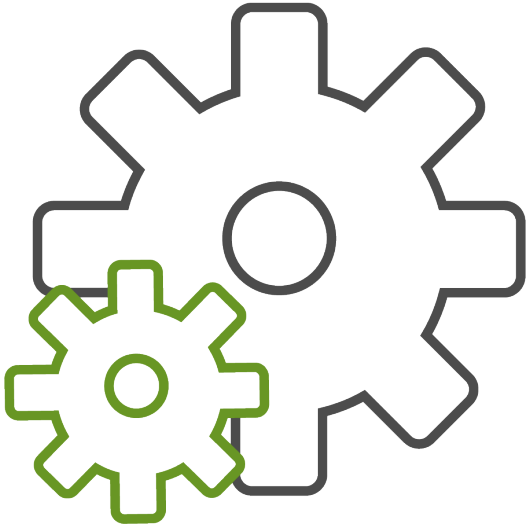
Credit: Vinod Khosla Keynote @ ONS 2014

kaloom

Ideally …

kaloom

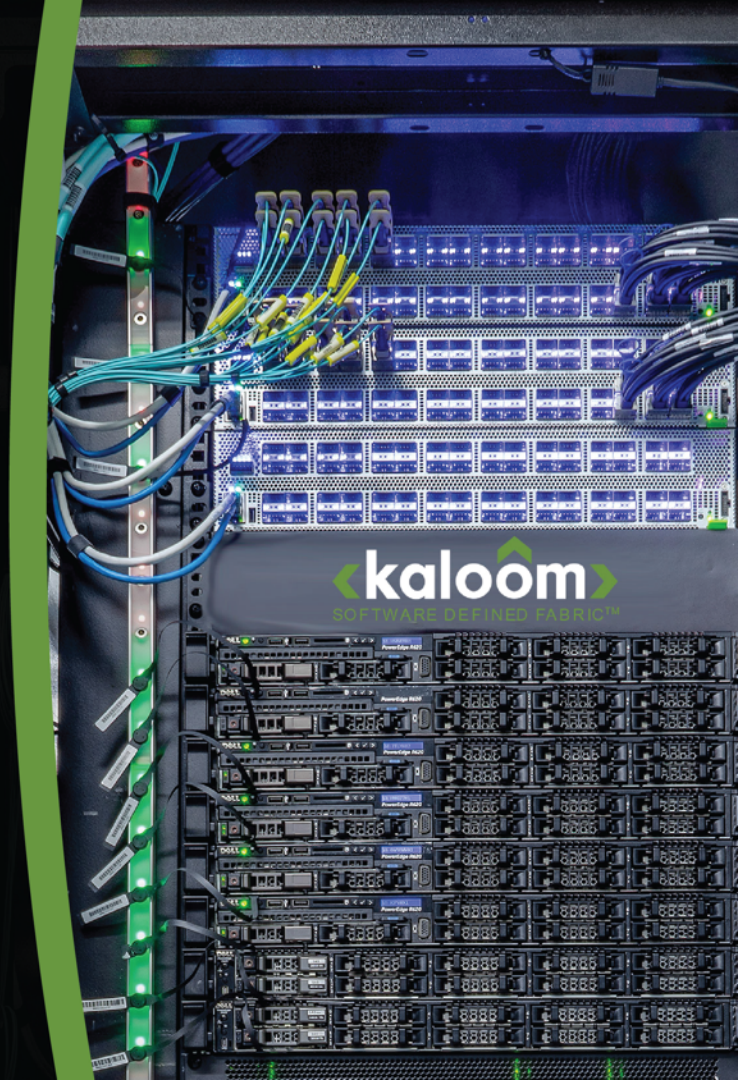
What does automation give you?



- **Savings** — Save time and money spent on provisioning
- **Time to deploy** — Save man hours spent on troubleshooting and fixing errors
- **Error Free** — Quicker time to deploy : Days → Minutes

Programmability

kaloom™



"Why does it take so long for networking equipment to add a new feature?"

- **Nick McKeown** Keynote Panel @ ONS 2018

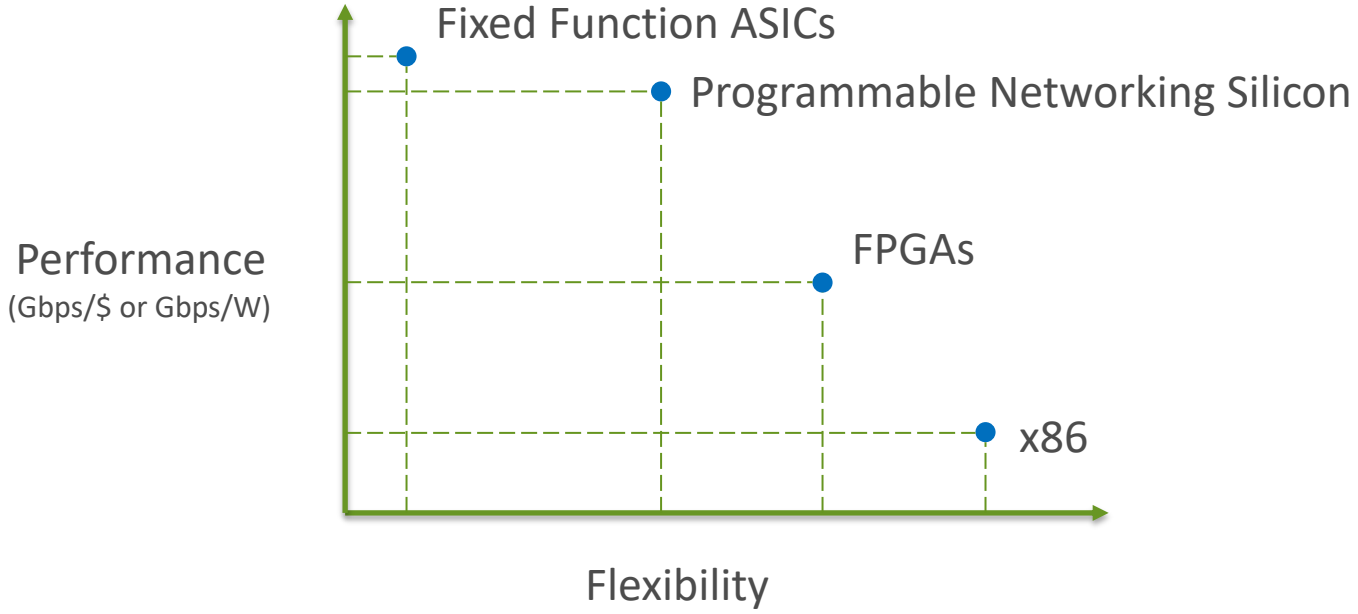
Programmability matters

- **Promotes feature velocity**
  - There is a huge time lag between widespread acceptance of a protocol and its appearance in networking silicon
- **Reduces/eliminates vendor lock-in**
  - No need to wait for a given vendor to provide a new chip or a new appliance
- **React quickly to protocol changes and fixes**
  - Over The Air updates just like compute

There is an obvious solution!!

Run everything on x86

… but there is no such thing as free lunch

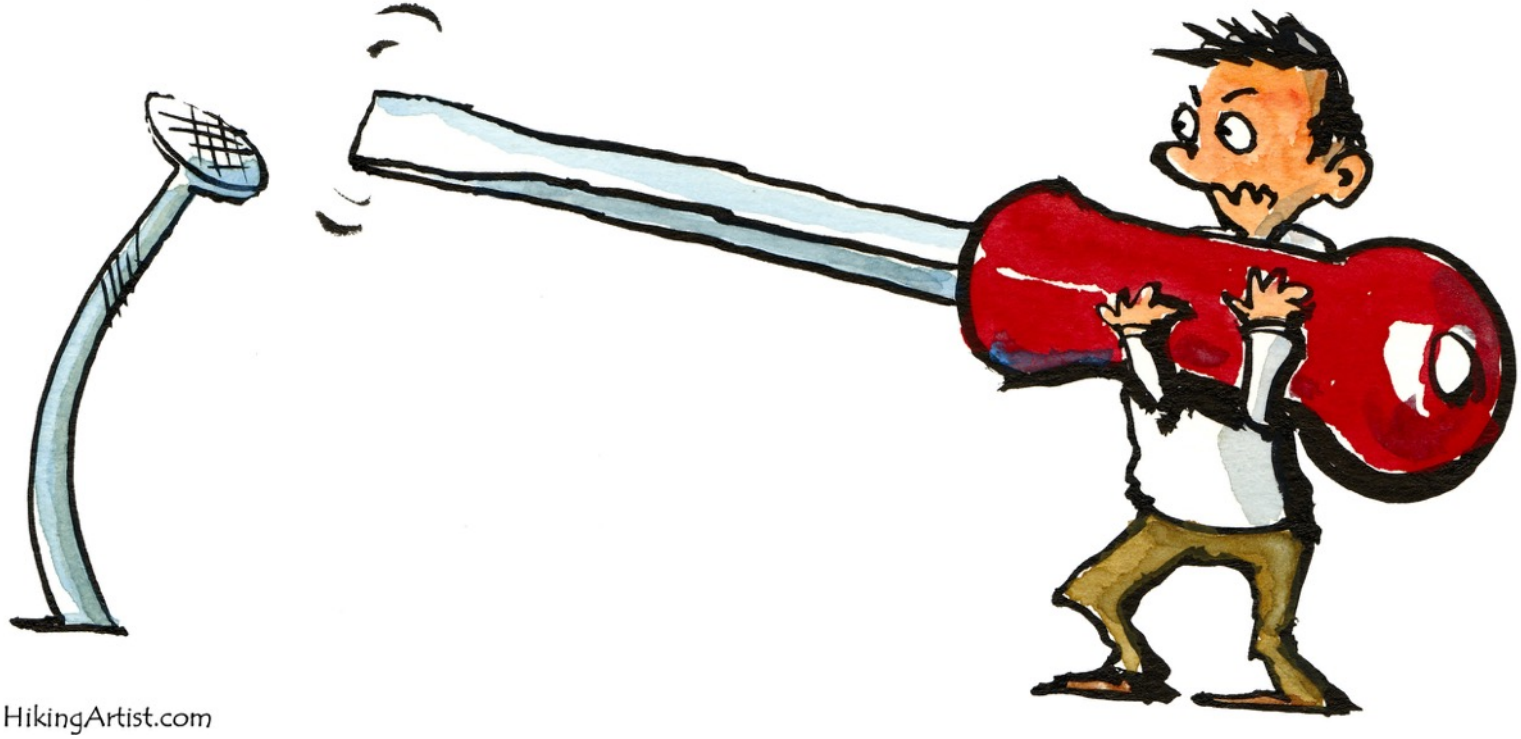


Cost and Power Usage comparisons

	Cost ($)	Throughput (Gbps)	$/Gbps	
Xeon Gold 6148 Server	~20000	400	50	32X
Barefoot Tofino 6.4 Tbps Switch	~10000	6400	1.5625	

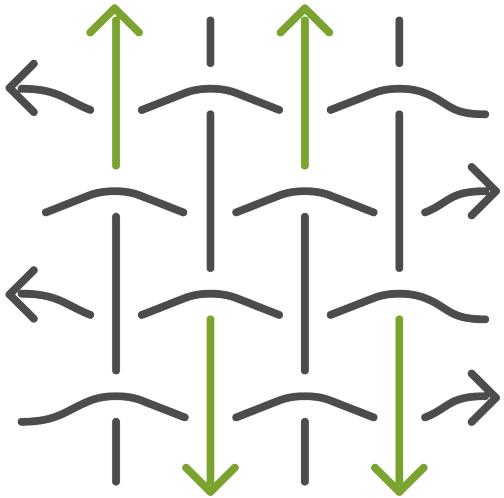
	Power (W)	Throughput (Gbps)	W/Gbps	
Xeon Gold 6148 Server	~1000	400	2.5	16X
Barefoot Tofino 6.4 Tbps Switch	~1000	6400	0.156	

The right tool for the right job



"Everything should be made as programmable as needed, but no more"

What does programmability give you?



- **Longer lifetime** — Hardware stays relevant for longer
- **Feature Velocity** — Rollout features faster
- **Convergence** — One Network to rule them all

kaloom

Containerization

kaloom™

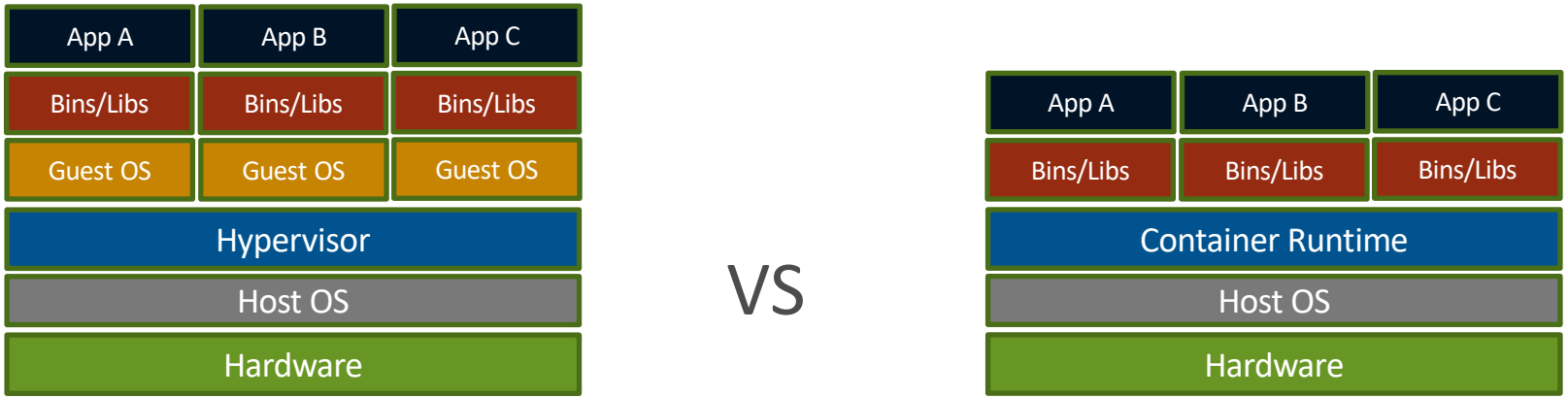
Containers are taking over

- Resource Efficient
- Deterministic
- Quick provisioning
- Modularity
- Scalable

kaloom

Resource Efficient

- Containers are simply more resource efficient than VMs
- They take up much less space on disk (MBs vs GBs)
- They consume way less memory while running (easily 100s of MBs per container)



Deterministic

- Containers can provide deterministic environments to deploy applications
  - Include all required dependencies
  - Isolated from other applications
- The consistency of containers allows deployment on varied environments
  - Deploy same application on premise and on multiple public clouds without worrying about the environment

Quick provisioning

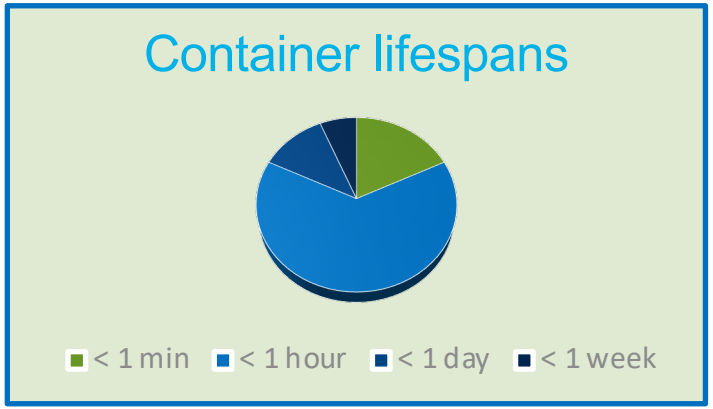
- Containers can be spun up and down in a matter of seconds
- Commonly used container orchestration systems support declarative management
  - E.g. Deploying something with 4 replicas would be as simple as stating

    ```
    replicas : 4
    ```

    in a configuration file.
  - The orchestration system then automatically takes care of ensuring this.

Scalable

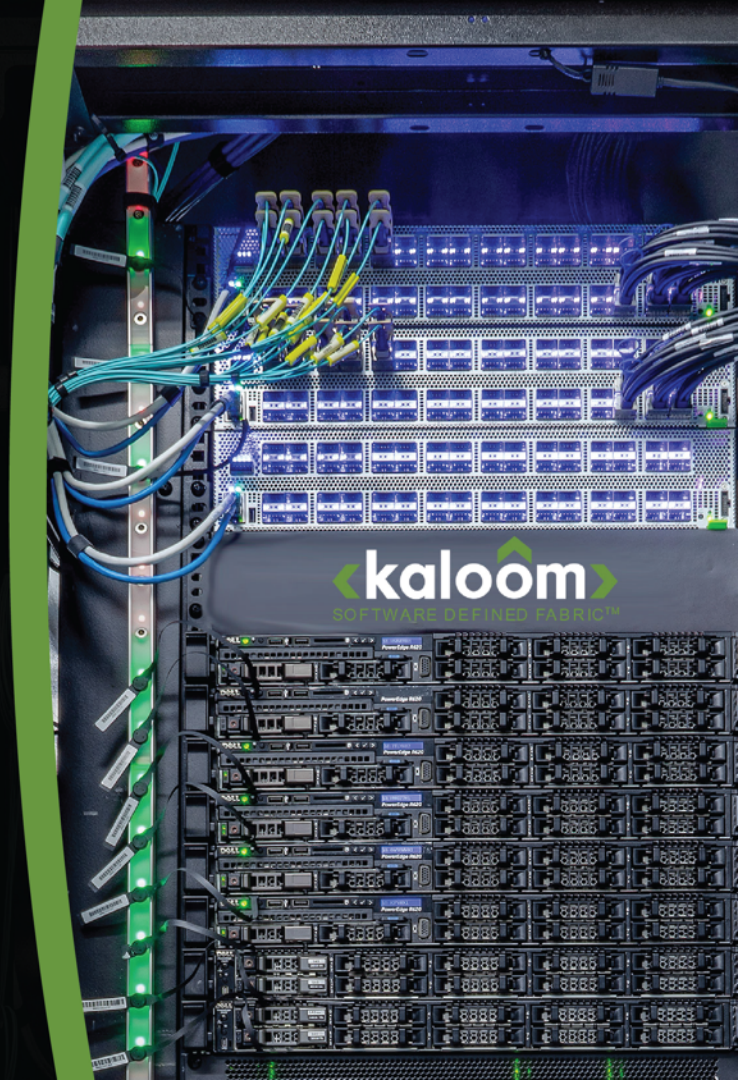
- Containers allow **independent** scaling of workloads
- A study by InfoWorld showed the following distribution of container lifespans
  - **17 percent less than a minute**
  - **78 percent less than an hour**
  - **89 percent less than a day**
  - **95 percent less than a week**
- Doing this using VMs would be impractical!



How does networking help?

- Containers work best with a microservice based architecture where pieces of the application are modularized and decoupled
- The microservices can be independently scaled and deployed across the datacenters
- They communicate using the datacenter fabric
- A datacenter fabric optimized for container networking can support container communication natively and efficiently
  - Provide support for container networking using an interface such as CNI (Container Network Interface)
  - Provide acceleration of container networking functions
  - Offload the servers from performing intensive networking functions

Integration



Why Integrate?

Reduce	Reduce complexity
Harmonize	Harmonize operations
Optimize	Optimize resources for performance and efficiency

Reduce	Reduce complexity

- Use the same kind of environment for networking as for compute and storage. i.e. use a standard operating system

Uniformity

- Same software orchestration model & management tools across networking, compute, and storage

Military-Grade Security

Uses a secure operating system that provides features that leverage:

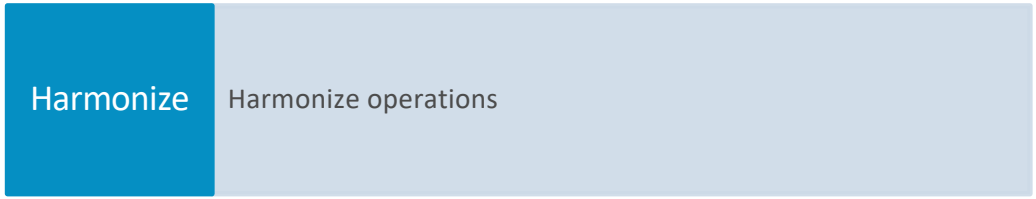
- SELinux
- cgroups
- Network namespaces

Rollback and Upgrade

- Easily rollback to a previous version of the OS
- Deploy updates in a single step

Minimal Footprint

- Lightweight OS that provides the flexible & modular capabilities of Linux containers



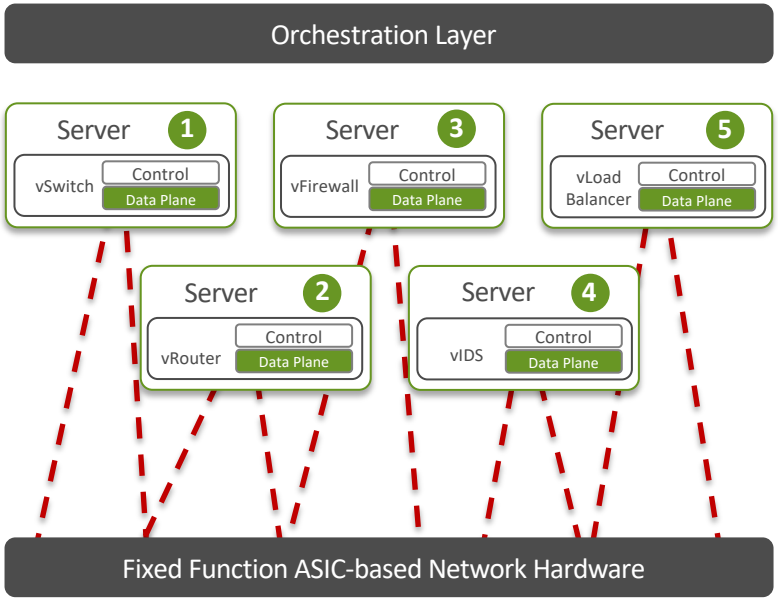
- Integrate network functions into the fabric as much as possible
  - This simplifies operations and improves efficiency
- Share the same infrastructure for multiple purposes. E.g. for testing and for production
  - Slice the datacenter to provide the required isolation
- Offload inefficient networking tasks away from the servers

Optimize

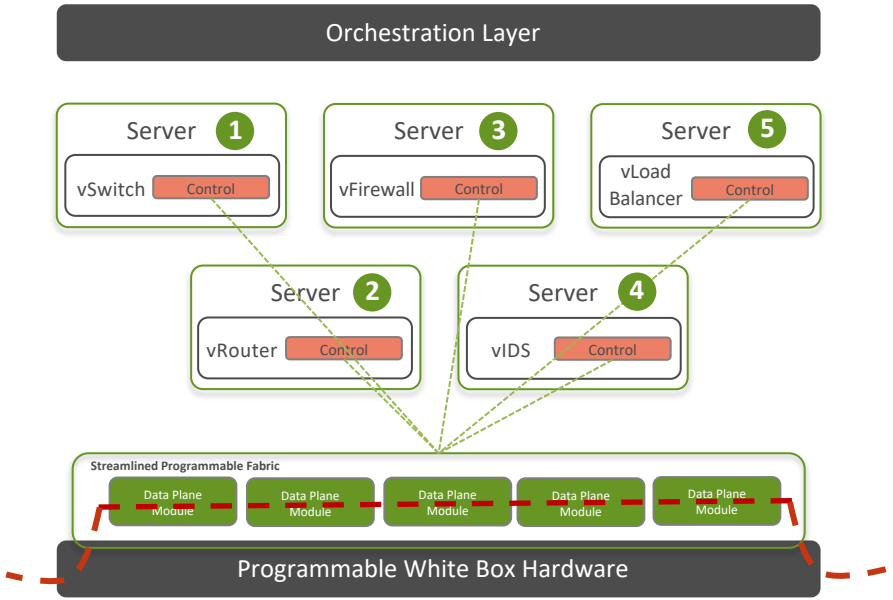
Optimize resources for performance and efficiency

- Provide the means to off-load L2-L7 networking functions from application and storage servers
- Allows hosting of L2-L7 stateful and stateless networking services within the fabric itself
- Reduces power consumption by more than 20% by integrating firewalling, load balancing, and other network functions
- Reduces power and latency by avoiding tromboning between application servers

Optimize Optimize resources for performance and efficiency

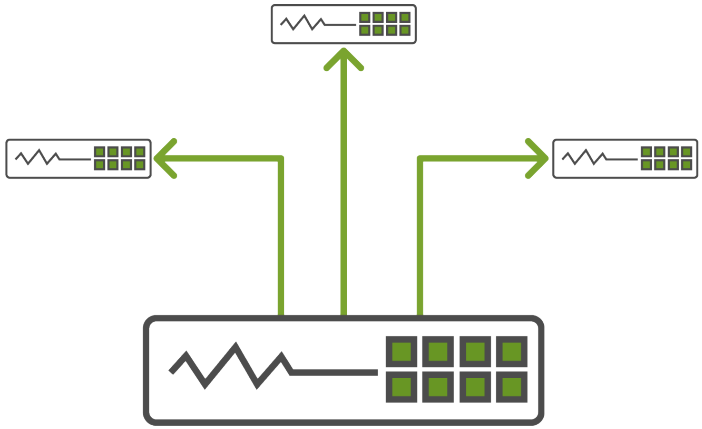


Without an optimized fabric



With an optimized fabric

What does integration give you?



Reduced Server Costs

Save on server costs

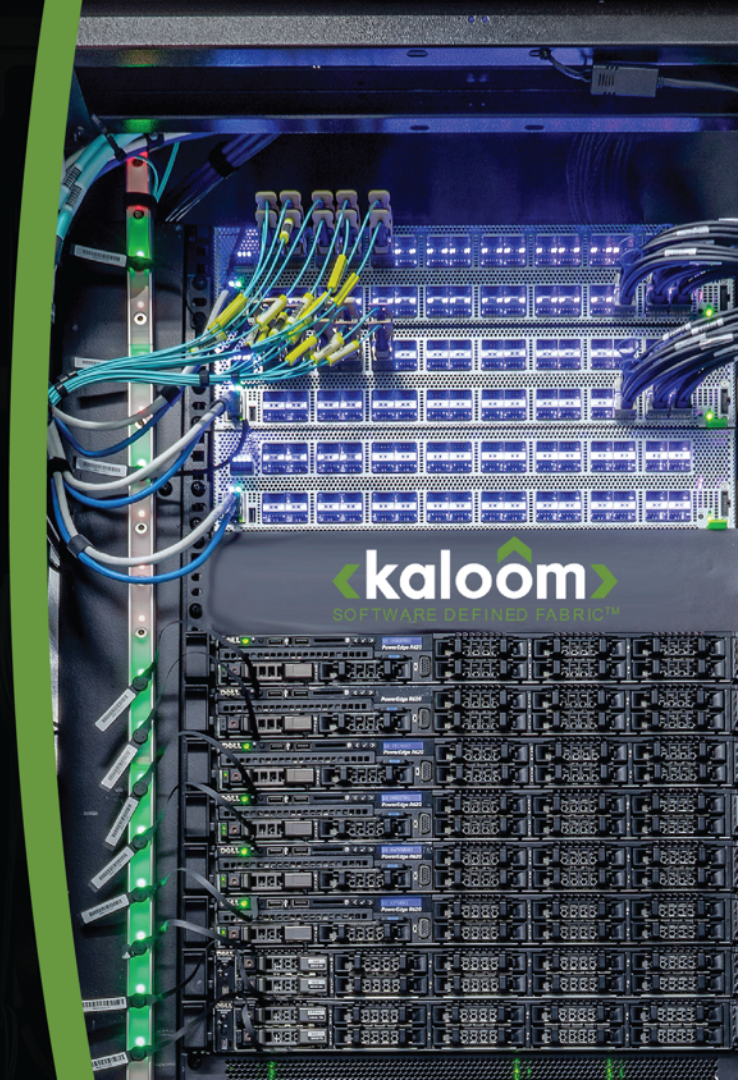
Reduced Complexity

Easier to manage fabric extensions than individual vSwitches or Smart NICs

Performance

Higher throughput and lower latency allows for newer apps

Summary

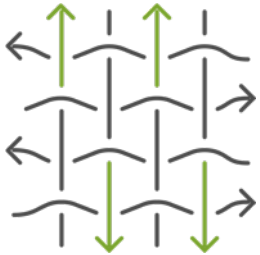


kaloom™

Summary

Automation

Containerization



Programmability

Integration

Questions?