eBPF-based Extensible Paravirtualization

Luigi Leonardi, Giuseppe Lettieri, Giacomo Pellicci

1. Paravirtualization
2. eBPF
3. Hyperup Calls
4. eBPF-based extensible Paravirtualization
5. Virtual to Physical CPU Affinity
6. Q&A

Paravirtualization

Possible Approaches

Hypercalls

Modify the guest kernel to give the hypervisor some hints

VM Introspection

The hypervisor analyzes the guest's memory

...

Hyperup calls

eBPF

- Useful debugging tool
- In Kernel Verifier
- Kernel is not modified

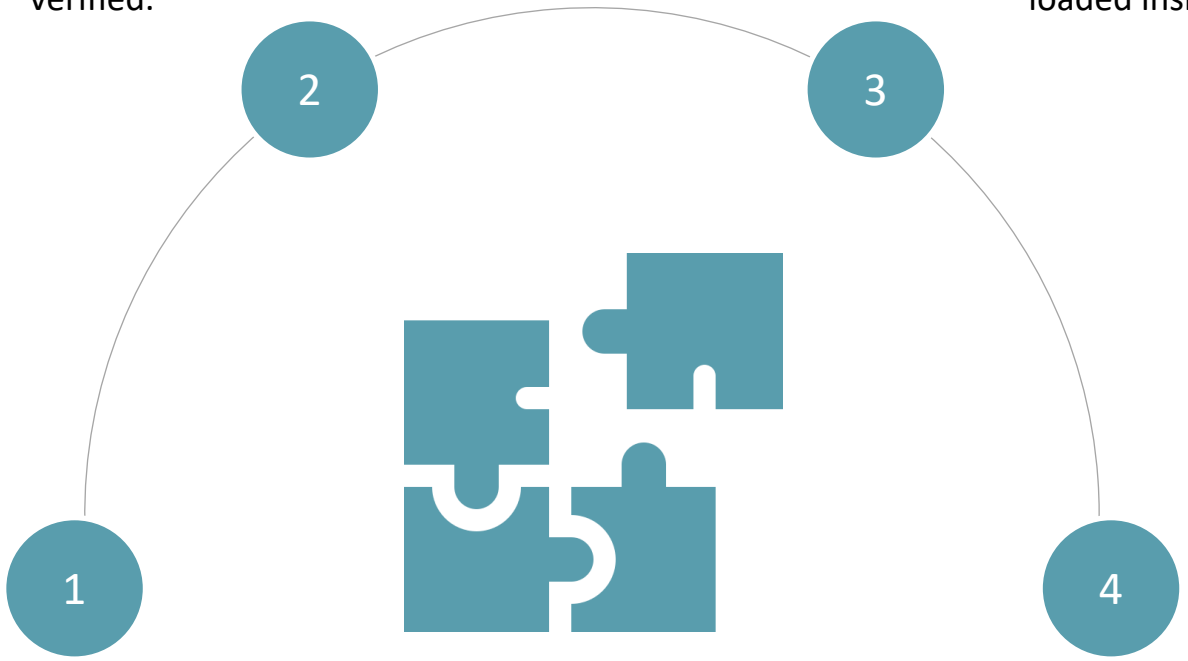
eBPF

Injection

Code is injected inside the Kernel, ready to be verified.

Verification

Injected code is verified and if successful is loaded inside the Kernel



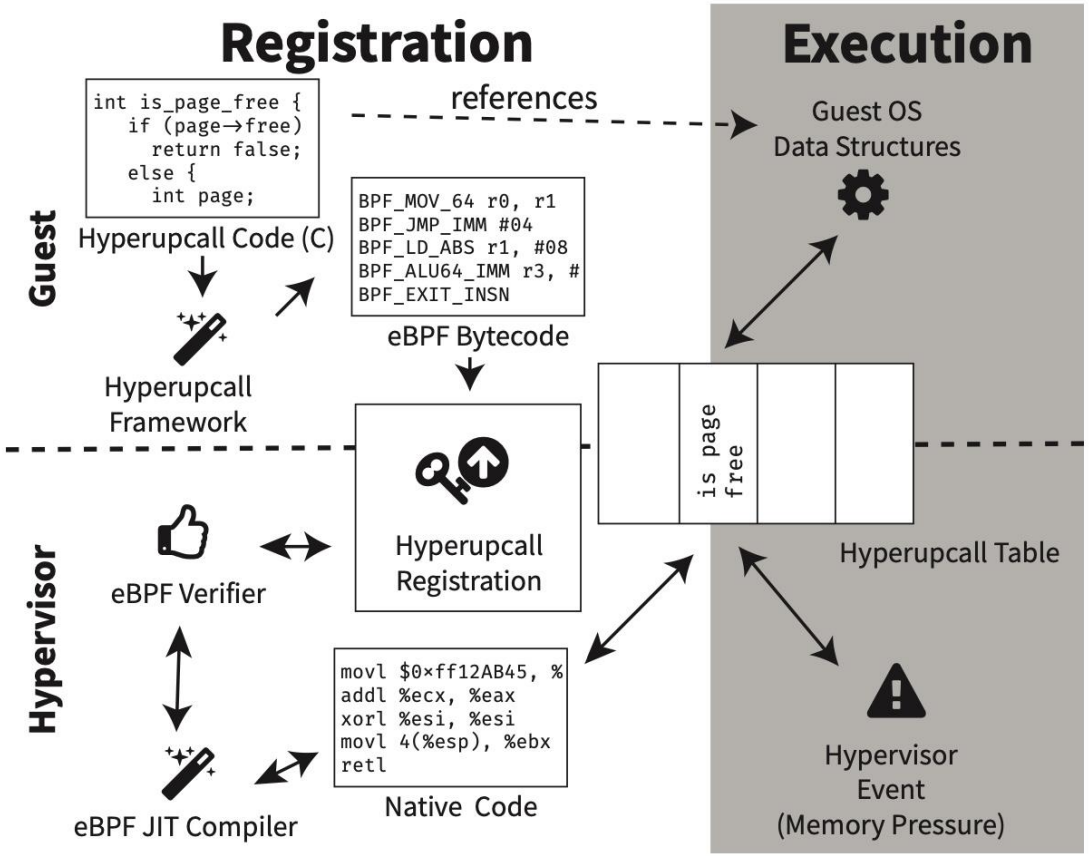
Generation

eBPF code is generated from C-like language and compiled into bytecode

Statistics

Stats or debugging infos can now be collected.

Hyperup Calls



Performed by the Hypervisor with low overhead

Code is verified before execution

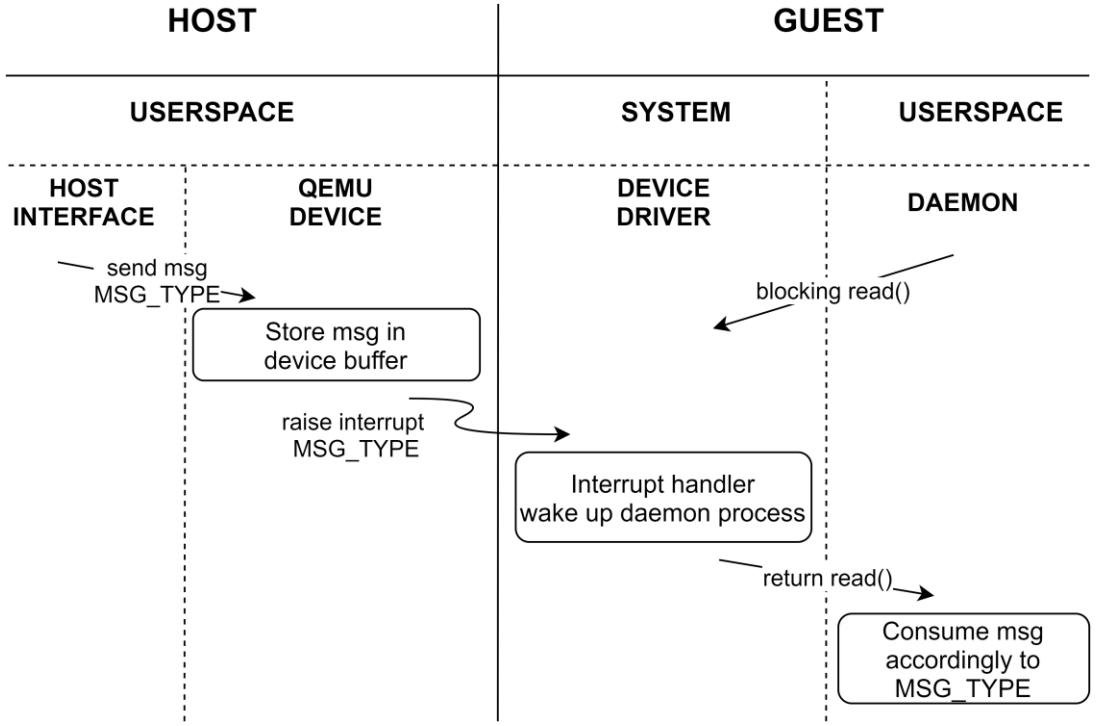
Ineffective when guest memory is encrypted

eBPF-based Extensible Paravirtualization

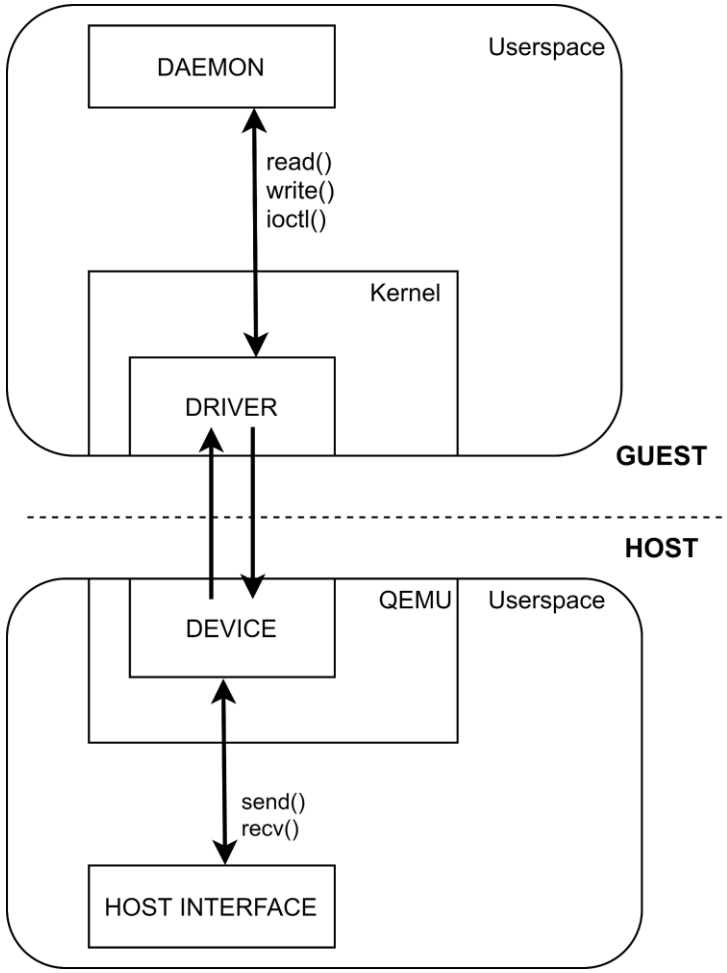
Host sends a message to the guest

Guest Agent consumes the message

eBPF code inside the message



eBPF-based Extensible Paravirtualization



Can be loaded or unloaded at any time

No need to modify the guest's Kernel

Guest is free to decide to load the eBPF program or not.

Comparison w.r.t. Hyperupcalls

eBPF-based Extensible Paravirtualization	Hyperupcalls
Host to Guest	Guest to Host
eBPF	eBPF
Async. Response	Invoked by the HV

Virtual to Physical CPUs Affinity

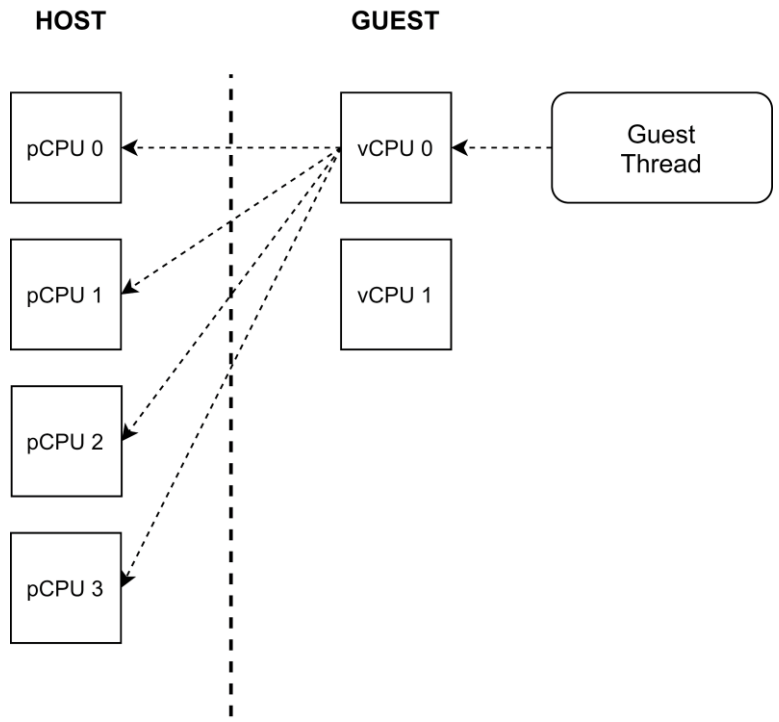
Reasons

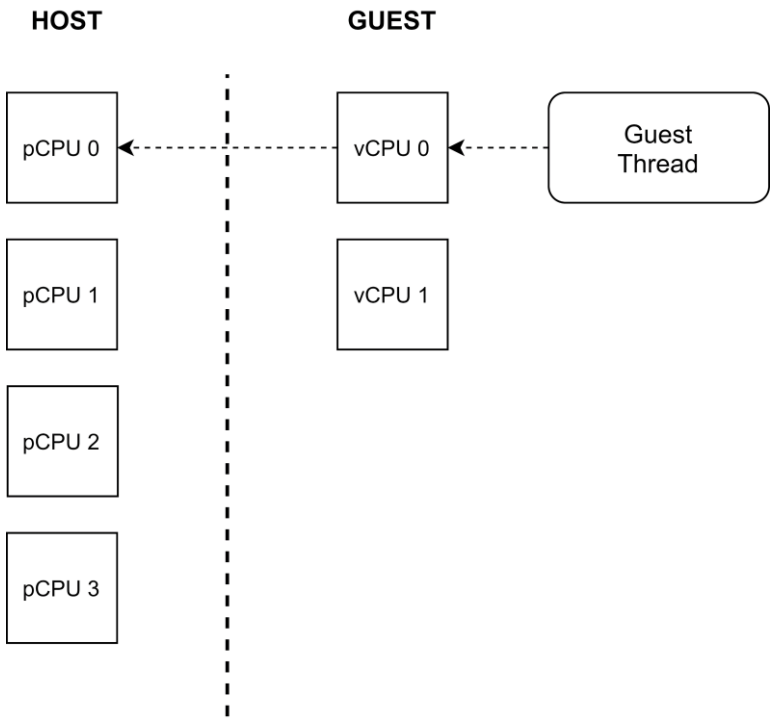
Speed

Flexibility

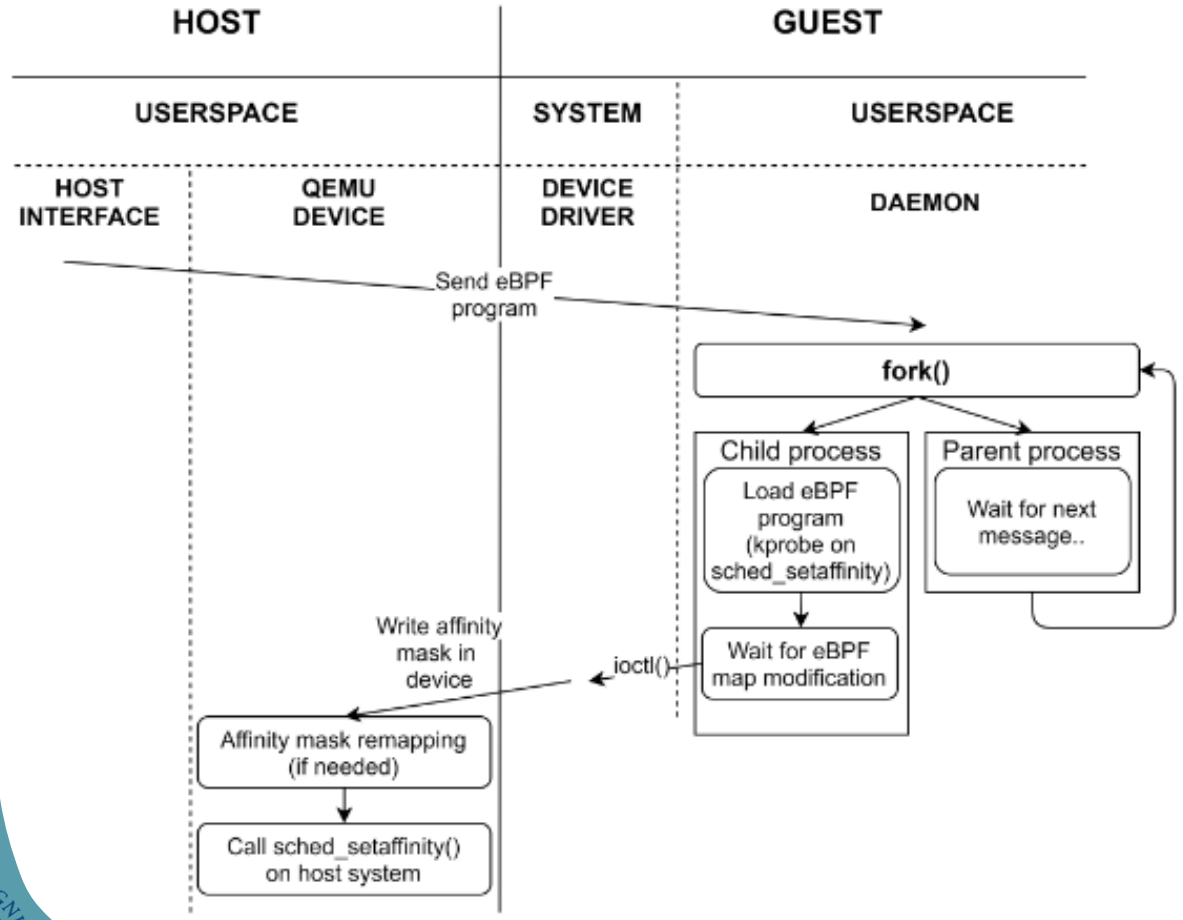
Security

Virtual to Physical CPUs Affinity





Virtual to Physical CPUs Affinity



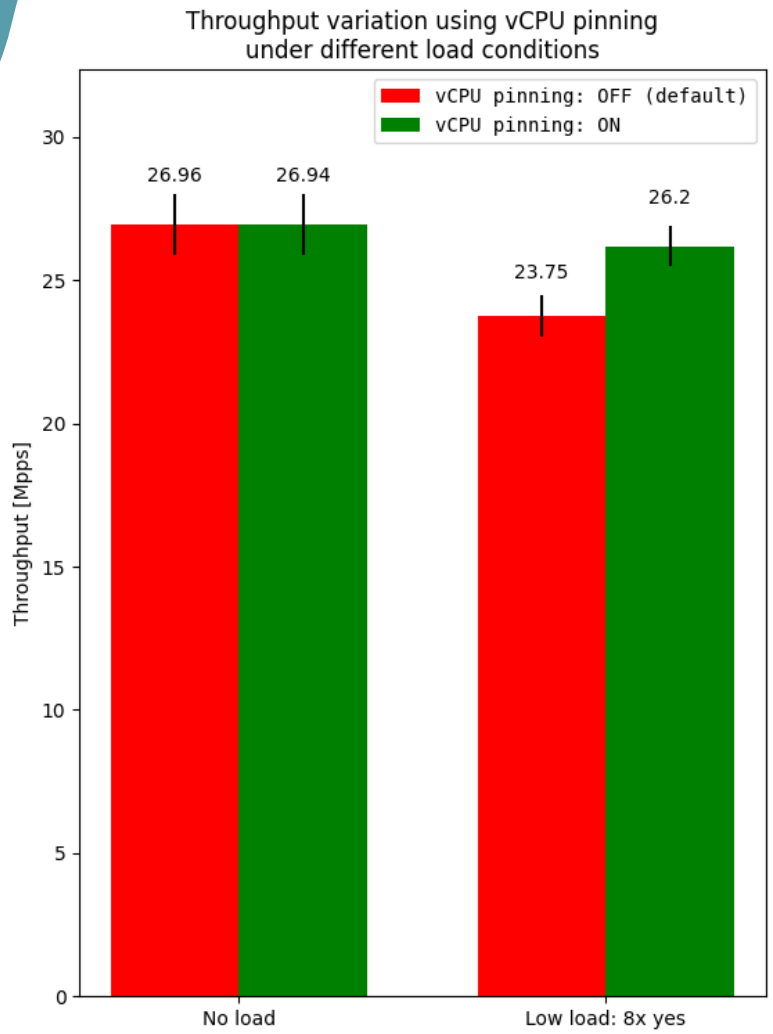
Kprobe on *sched_settaffinity*

Guest Agent checks for changes in eBPF map

Information on bindings is sent to the hypervisor

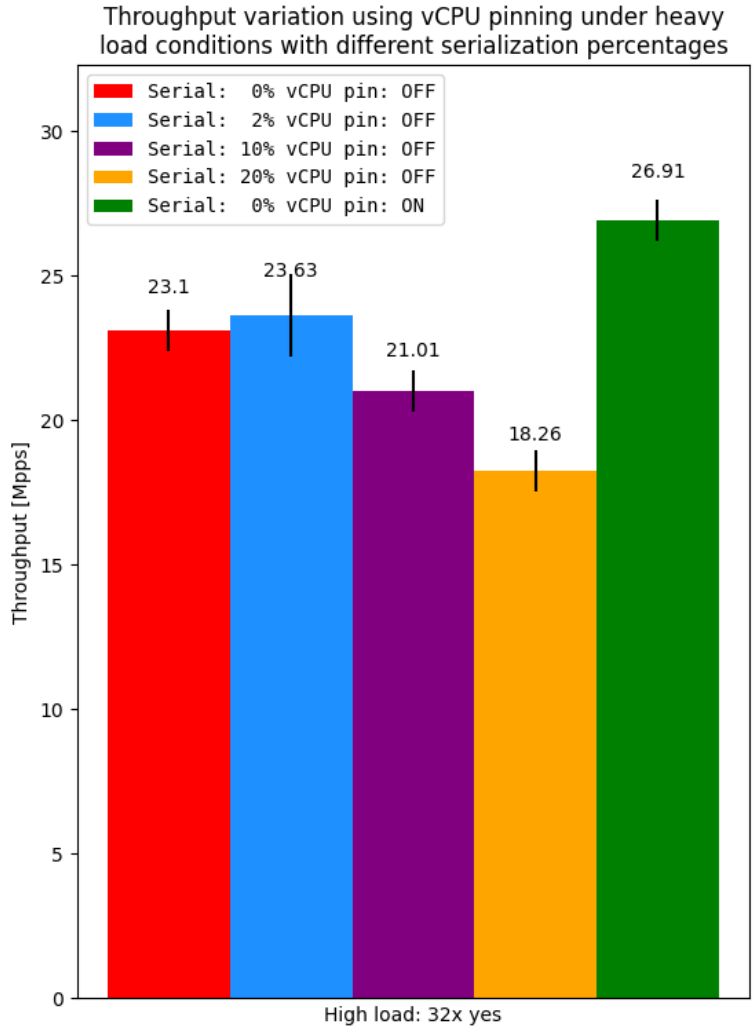
Performance Evaluation

vCPU Pinning w/o HT



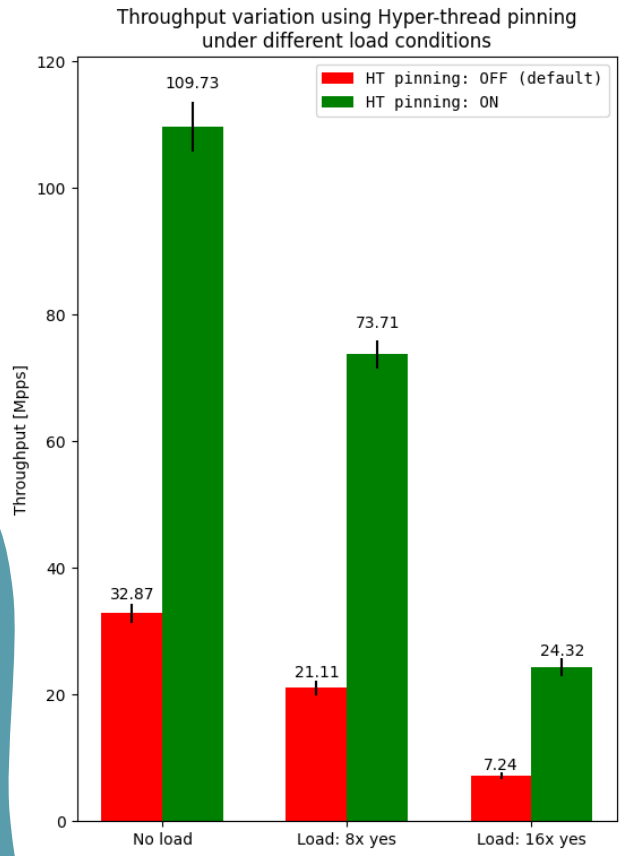
No Load
Low Load

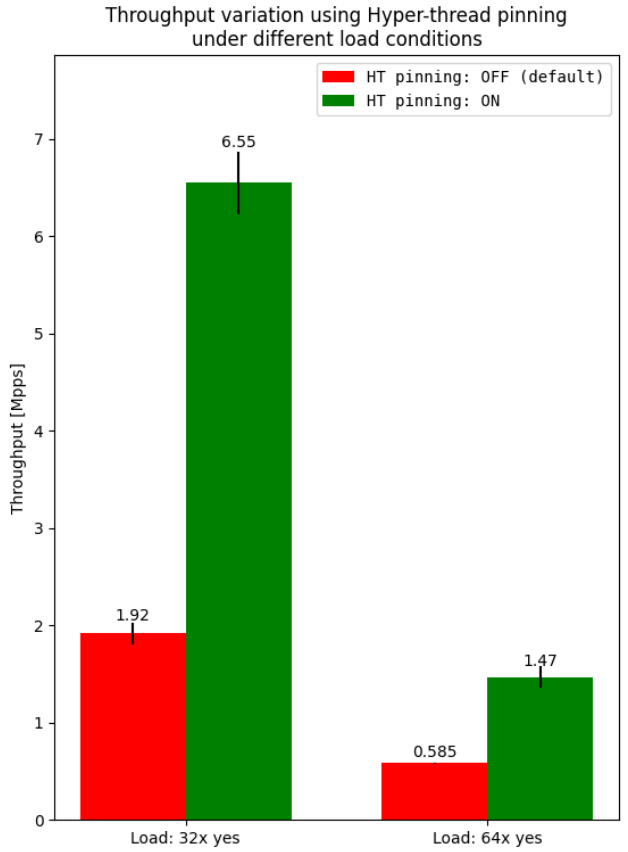
Serialization



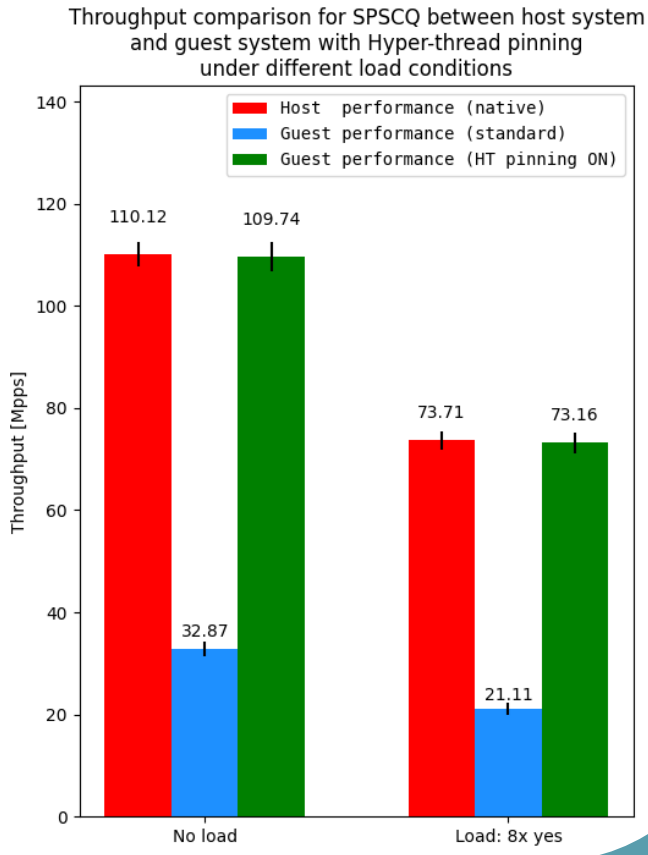
Performance Evaluation

vCPU Pinning with HT





Native Perfomance



No Load
Low Load

High Load

Thank You!

Any Questions?