Low latency edge computing with QEMU/KVM: Challenges and future

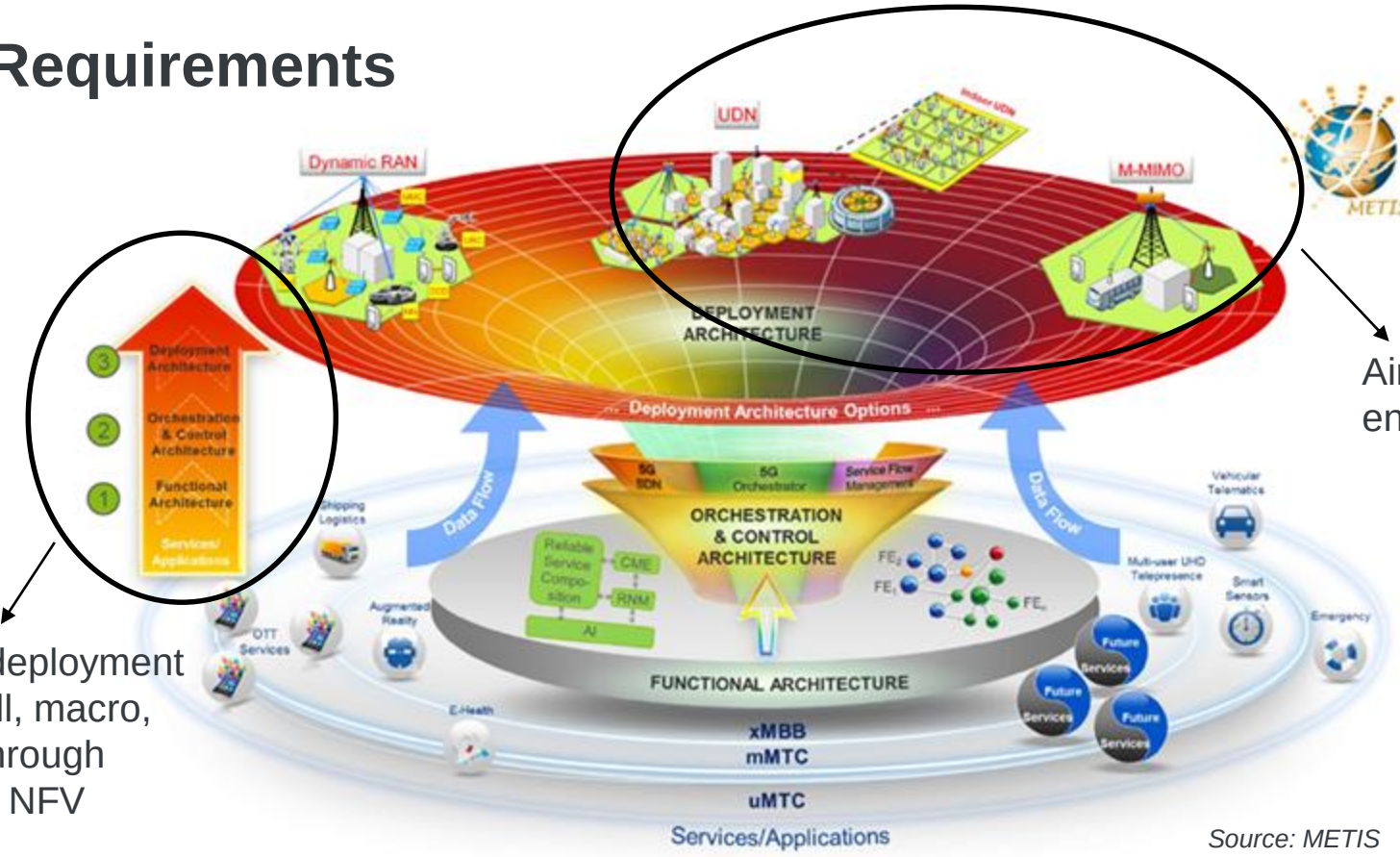
Mihai Caraman, PhD | Virtualization Architect

August, 2015

Agenda

- 5G Requirements
  - Base Station Virtualization
- RT KVM
  - Embedded PA and ARMv8
- I/O Virtualization
  - Direct Assignment
  - Virtio
- vSwitch Offload
- Open Stack Integration
- Tuning & Performance
- Future

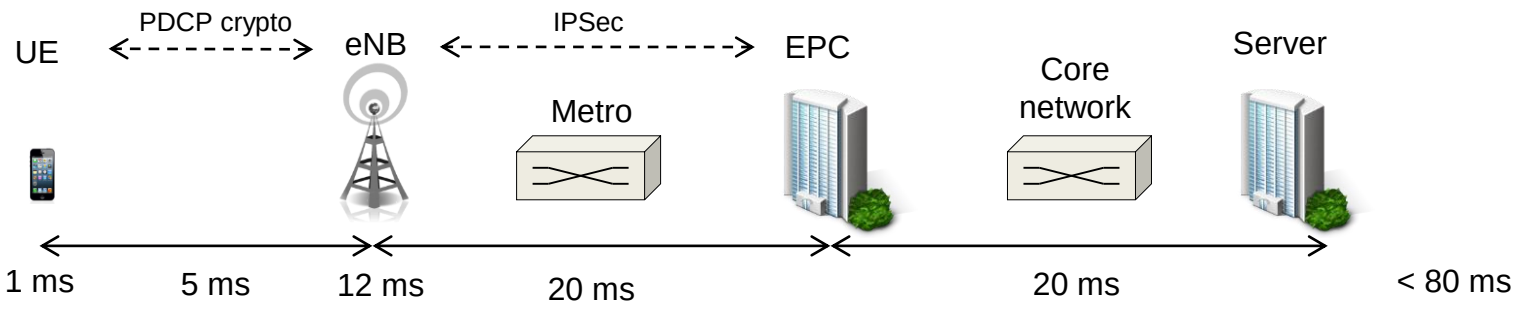
5G Requirements



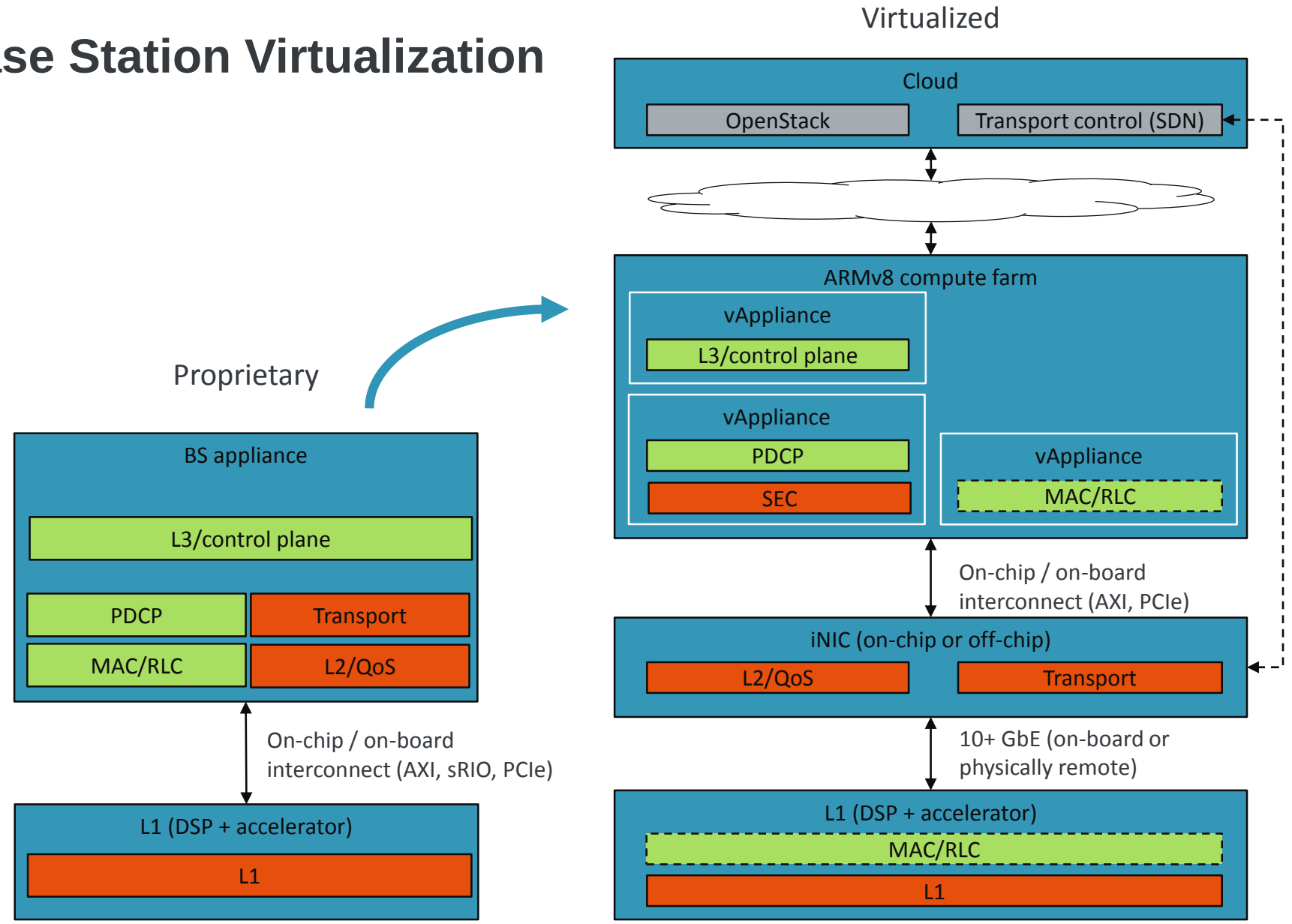
Air interface enhancements

Flexible deployment (small cell, macro, CRAN) through SDN and NFV

Source: METIS



Base Station Virtualization



Virtual Base Station - PoC

- Scenarios
  - L2/L3 stack in VM, end-to-end video download using commercial LTE dongle
  - PDCP scaling

Platform Phase I

- QorIQ T4240
  - PA Book III-E
  - Security Engine
- QorIQ T4240 PCI SR-IOV intelligent NIC

Platform Phase II

- QorIQ LS2085A
  - ARMv8
  - DPAA2 Devices with Management Complex (MC) configuration bus
  - Advanced I/O Processor (AIOP) integrated NPU

Virtualization requirements

Timing, latency requirements

- Transmission Time Interval (TTI)
  - Synchronized between L2 and L1 at 1 ms
  - Provisioned through GPS, IEEE1588/PTP (interrupt for demo purposes)
  - L1 with a 1Gbps interface adds 150us latency, 10Gbps: 15us
  - TTI IRQ delivered to guest user space application ~50us max latency

KVM

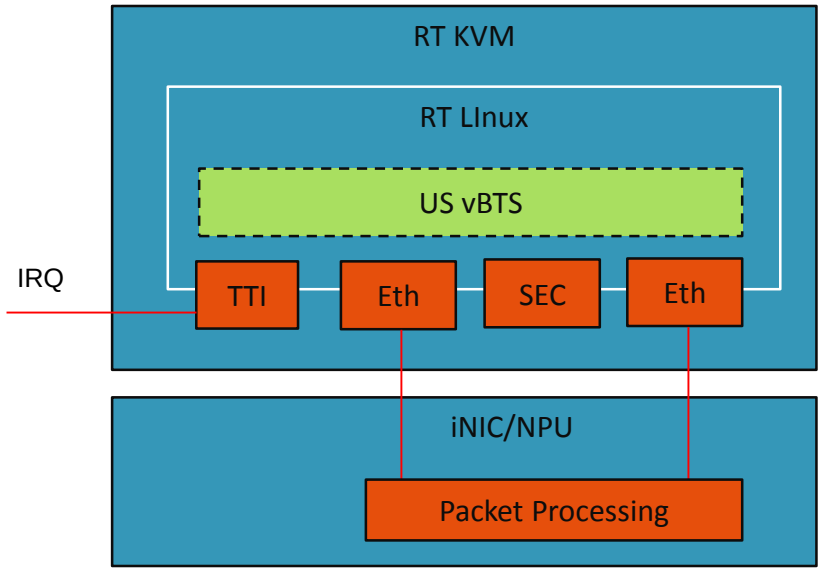
- RT Linux guest
- One RT vcpu per cpu
- CPU oversubscription (nice to have)
- TTI IRQ

I/O Virtualization

- Direct assignment
- Virtio

Open Stack Integration

Latency breakdown	
SoC infrastructure + driver	5 µS
PCIe	3 µS (150 Mbps)
SoC infrastructure	1 µS
Soft-switch	10 µS
Ethernet Tx	0.5 µS
Physical 10GbE	15 µS (150Mbps)
Switch	0.2 µS
Physical 1GbE	150 µS (150Mbps)
Ethernet Rx (IRQ)	30 µS
SoC infrastructure	1 µS ?
Driver	10 µS
Aggregate	<200 µS



RT KVM - PA Book III-E

MPIC emulation

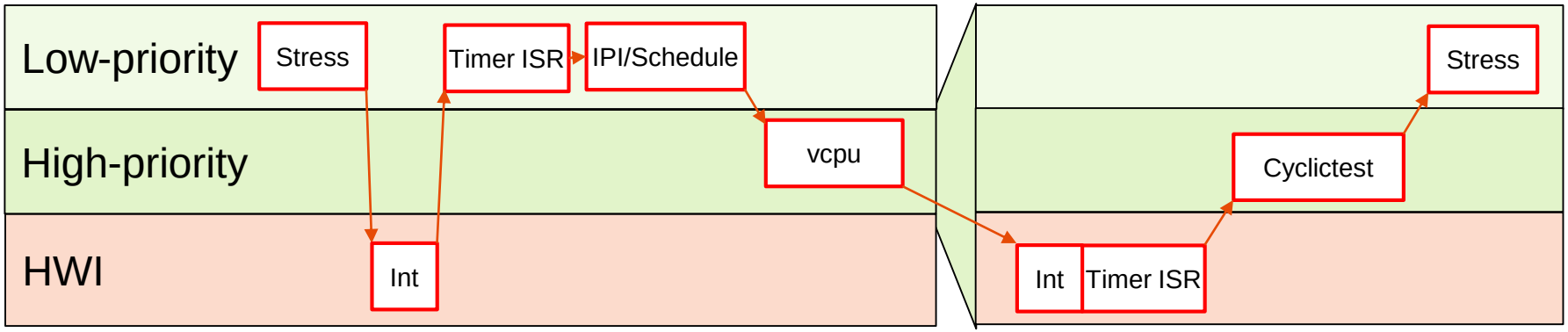
- Replaced spinlocks with raw spinlocks. PREEMPT_RT spinlock implementation uses sleeping mutex
- RT-friendly refactoring (to do):
  - Increase lock granularity
  - Minimize interrupts disabled code paths
  - Avoid races with (lazy) pending exceptions

Timer

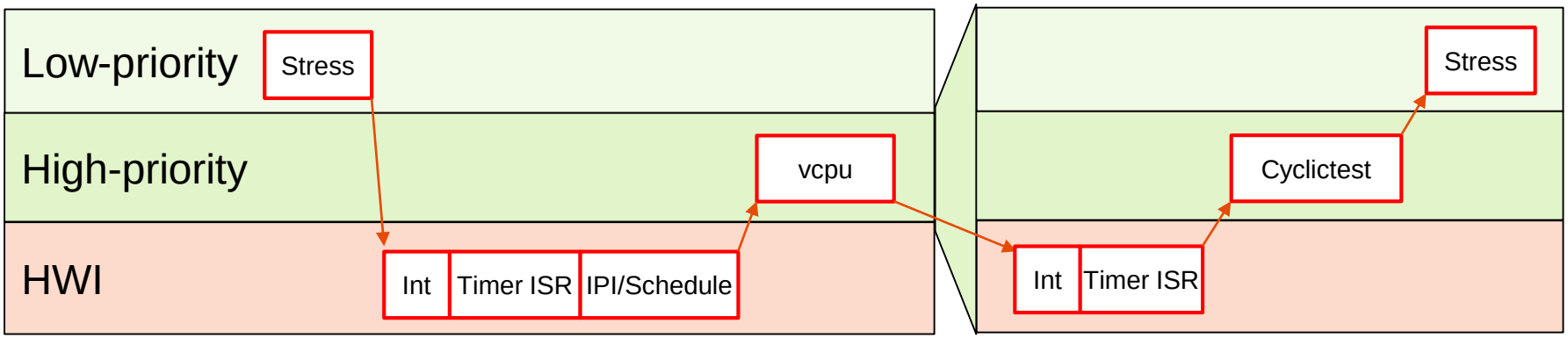
- Moved the processing of the KVM decrementer from softirq (ksoftirqd kthread) to hardirq context
- Replaced wait queues with simple work queues, wait queue callbacks prevent the use of raw spinlocks
- Removed unnecessary tasklet used by the hrtimer

RT KVM - Timer Interrupt Latency

Initial



RT



RT KVM - TTI IRQ & Latency Tracer

TTI IRQ

- GPIO assigned
  - Fast path delivery
- GPIO interrupt affinity
- Extensive host and guest debug statistics

Latency Tracer

- The tracer is a mix between 2 of the available ftrace modes of operation:
  - function-trace / function-graph
  - max latency - retain the maximum latency of this execution chain
- Expose the maximum execution latency for injecting the TTI interrupt into the guest and the associated code path.
- Used to analyze the causes for the TTI interrupt delivery latency

RT KVM - Configuration

```
# defconfig
CONFIG_RCU_NOCB_CPU=y
CONFIG_RCU_CPU_STALL_TIMEOUT=300
# CONFIG_MMC is not set

# bootargs
isolcpus=17-23 rcu_nocbs=17-23

# disable RT throttling
echo -1 >/proc/sys/kernel/sched_rt_runtime_us
echo -1 >/proc/sys/kernel/sched_rt_period_us

# disable RCU stall warnings
echo 1 > /sys/module/rcupdate/parameters/rcu_cpu_stall_suppress
echo 0 > /sys/module/rcupdate/parameters/rcu_cpu_stall_timeout

#  taskset & chrt
taskset -c $CPU_QEMU qemu-system-x …; chrt -p 95 $QEMUPID
taskset -pc $CPU_VCPU $VCPUPID; chrt -p 95 $VCPUPID
```

RT KVM - PV-SCHED

Para-virtualized scheduling porting on PA Book III-E

- Follows the original implementation of Jan Kiszka on x86, adapted for PA and re-based on newer kernels
- The significant differences are interrupt handling and delivery to the guest

I/O Virtualization - Direct Assignment

PCI SR-IOV VFs direct assignment (VFIO PCI)

- Place VFs in different IOMMU groups
  - Access Control Services quirks
- PCI EP partitioning PoC
  - Device ID allocation and programming
  - Enable IOMMU entries after each device attach
  - Dump IOMMU information for debugging
- Arch fix-ups are evil
  - QEMU PCI EP inadvertently hidden
- Memory translation
- MSI interrupt affinity

PCI DMA memory translations:

T4-iNIC:iNIC app (receives physical/iova addr from driver)
|
T4-iNIC:DMA engine
|
T4-iNIC:outbound PCI ATMU
|
PCI bus (bus:dev:func)
|
T4-RDB:inbound PCI ATMU
|
T4-RDB:PCI controller maps bus:dev:func to LIODN
|
T4-RDB:PAMU (iova to real addr)
|
T4-RDB:DDR

I/O Virtualization - Direct Assignment

Security engine direct assignment (VFIO for Platform Devices)

- QEMU glue code
- Physical and virtual functions

Management Complex Bus device assignment (VFIO MC)

- VFIO for Management Complex
  - QEMU integration
  - Legacy interrupts with irqfd support
  - Performance improvements
    - KVM ARM support for direct I/O Guest caching attributes (I/O portal)
    - I/O portal HW interrupt coalescing

I/O Virtualization - virtio-crypto

Provide binary compatibility across HW platforms (from different vendors) and machine migration

Virtio-crypto device

- supplies cryptographic offloading for the guest

Cryptographic transformations:

- ablkcipher - block ciphers (encryption)
- ahash - hashing (authentication)
- aead - authentication and encryption in one job

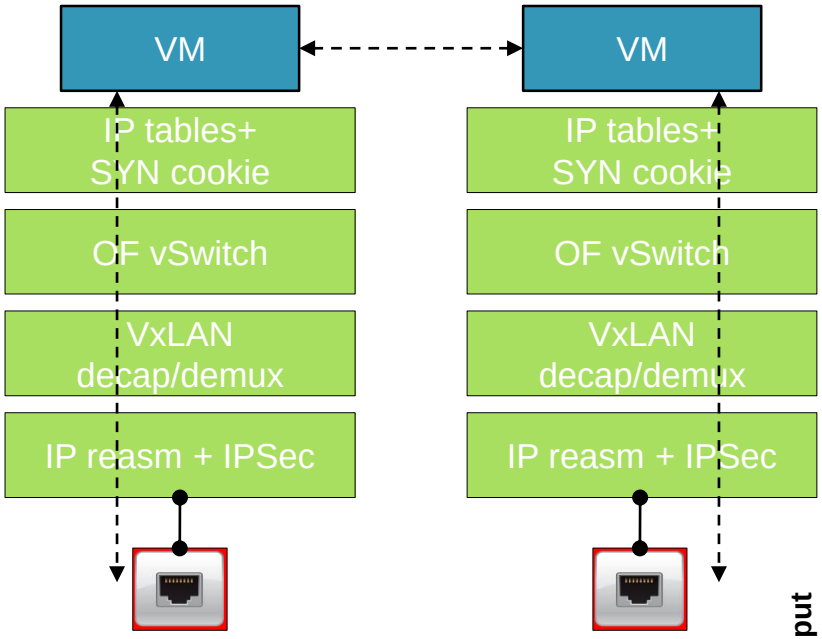
Virtqueues

- session, crypto, control

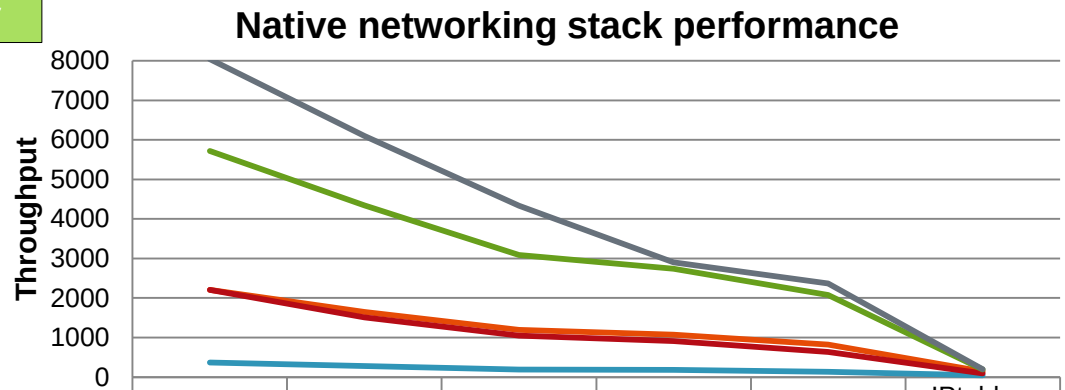
PoC

- QEMU integration
- Linux frontend driver
- vhost-crypto over crypto-API

Packet Processing Performance



- Increasing complexity of infrastructure stack
- Performance bottleneck from software implementation of networking stack

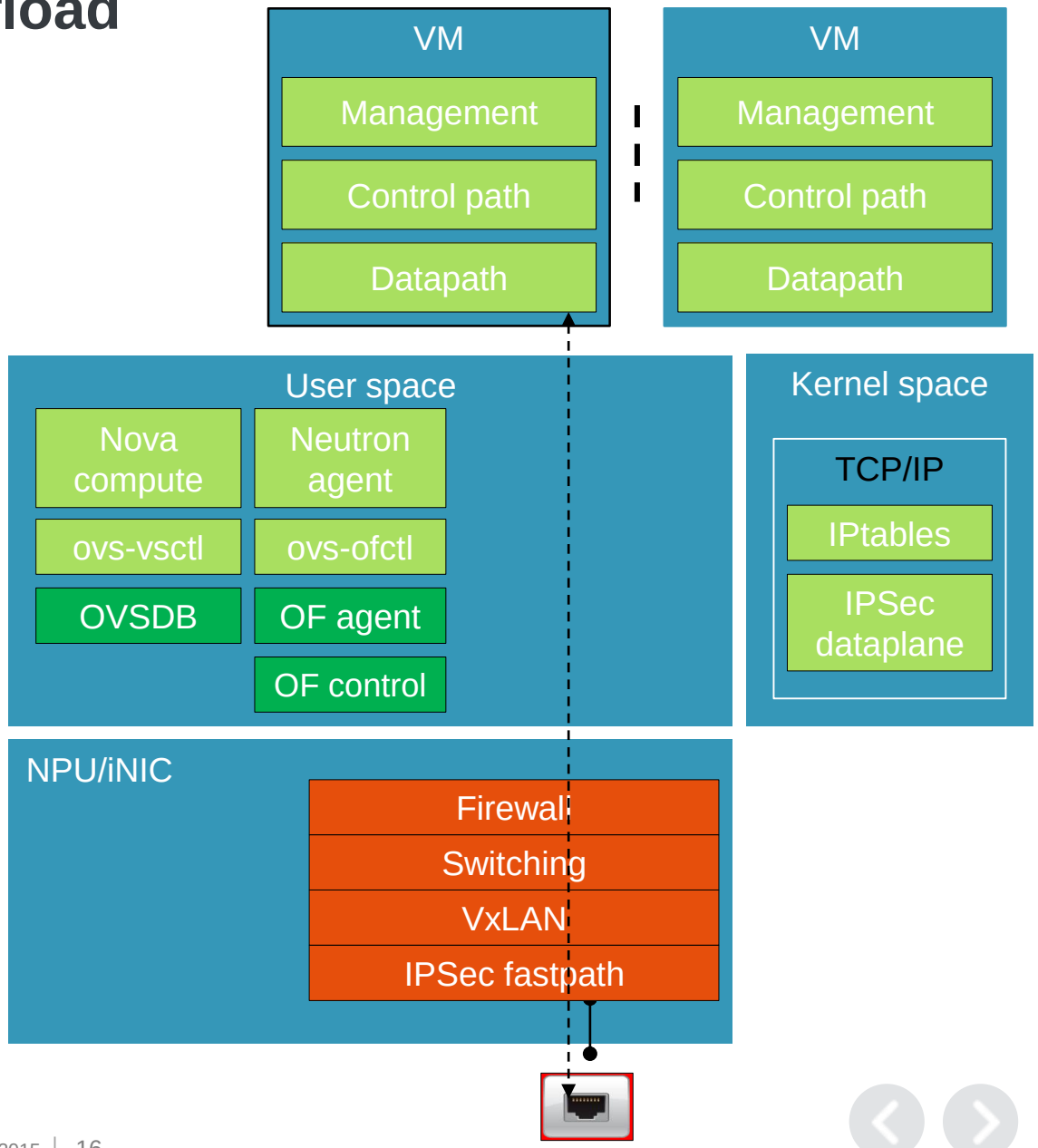


	TCP/IP	OVS	IPtables + OVS	OVS + VxLAN	IPtables + OVS + VxLAN	IPtables + OVS + VxLAN + IPsec
64	370	279	195	181	136	49
390	2205	1652	1194	1072	824	146
390 (1K conn)	2205	1514	1051	914	639	98
1024	5722	4346	3085	2737	2080	190
1472	8042	6097	4334	2906	2365	197

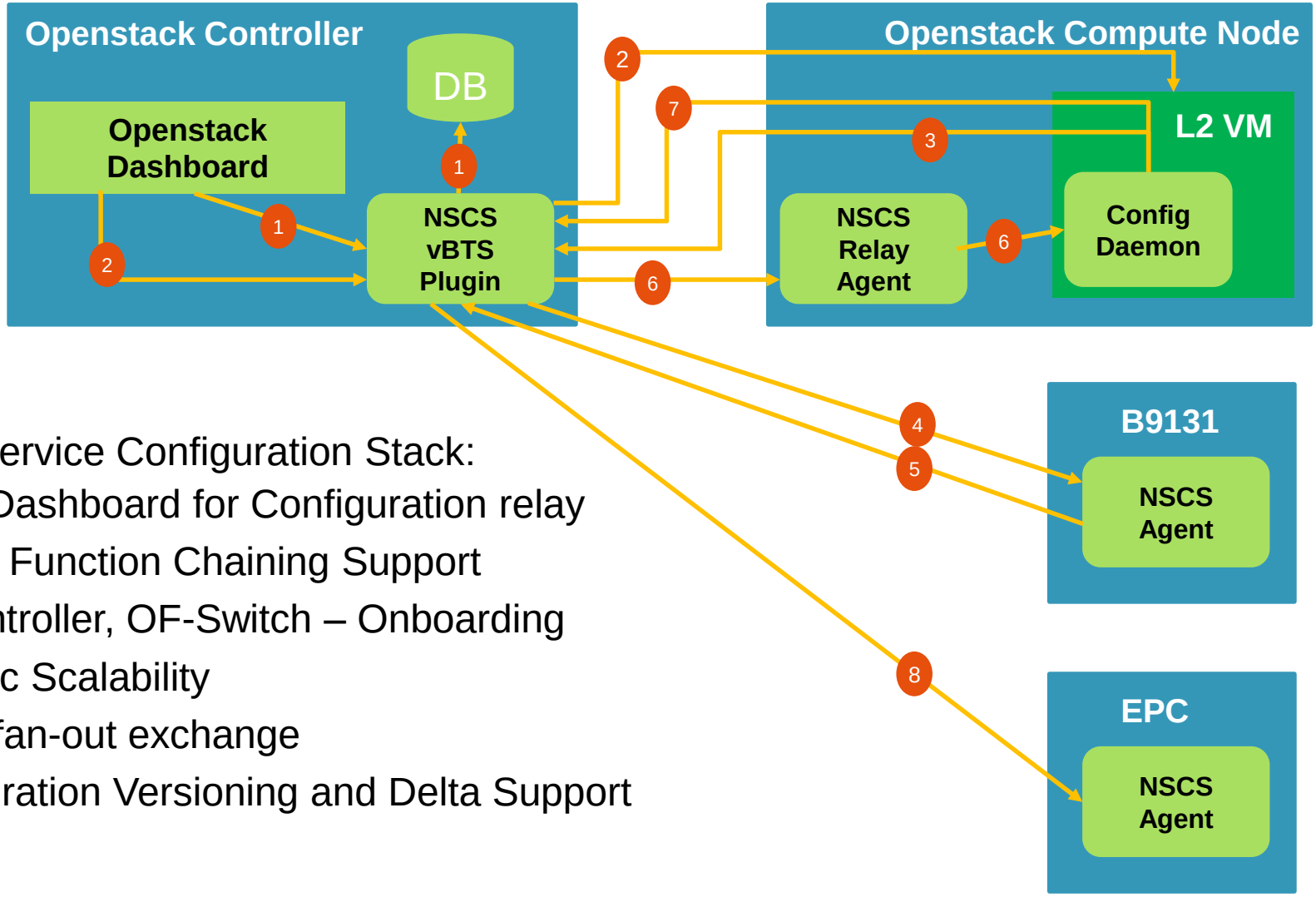
freescale™

Packet Processing Offload

- Limited GPP involvement (management/CP only)
- Offload as much packet processing to NPU/iNIC
  - NPU implements fast path
  - Direct connectivity to VM
- Faster Connection rate
  - IP Table Policy Caching
  - Entire OF pipeline processing for switching
  - All OF based data paths



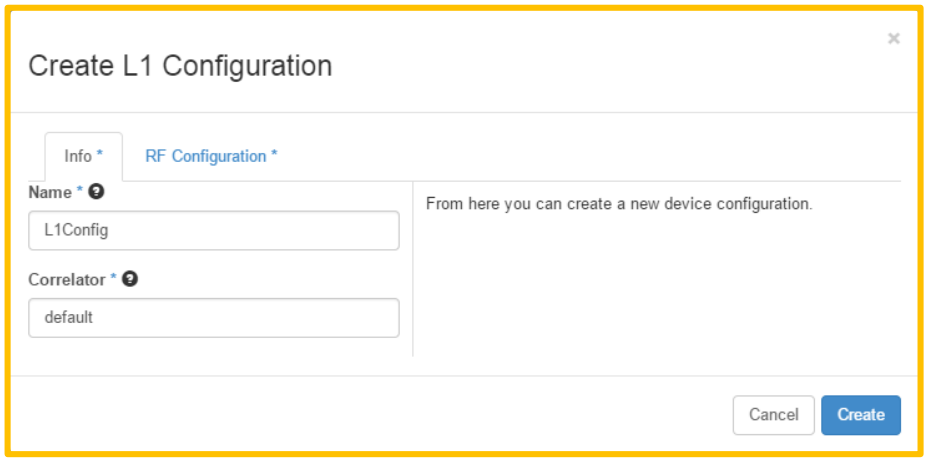
Open Stack - Integration

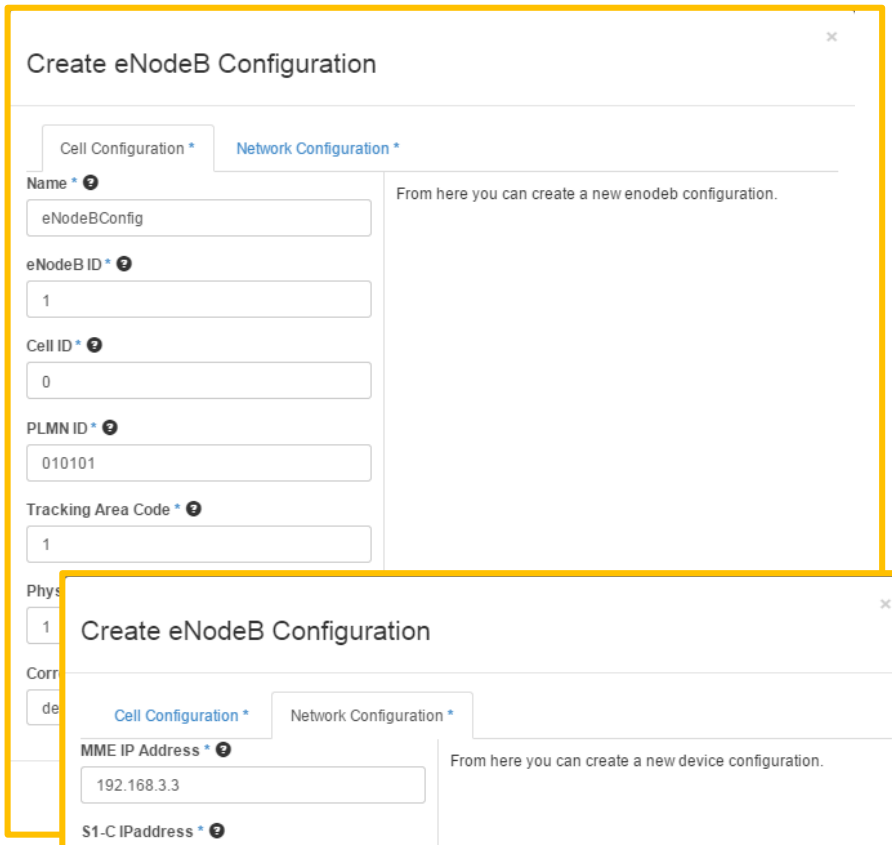


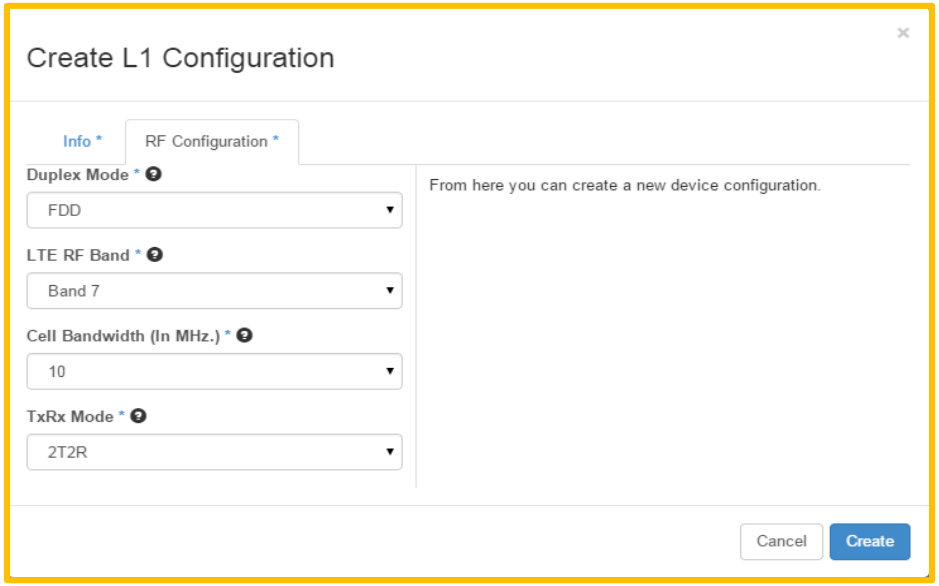
Network Service Configuration Stack:

- Single Dashboard for Configuration relay
- Service Function Chaining Support
- OF-Controller, OF-Switch – Onboarding
- Dynamic Scalability
- AMQP fan-out exchange
- Configuration Versioning and Delta Support

OpenStack - L1 & eNodeB integration







Tuning & performance

Latency Benchmarks

- Cyclictest
  - Stress: coremark, lmbench
- L2 application

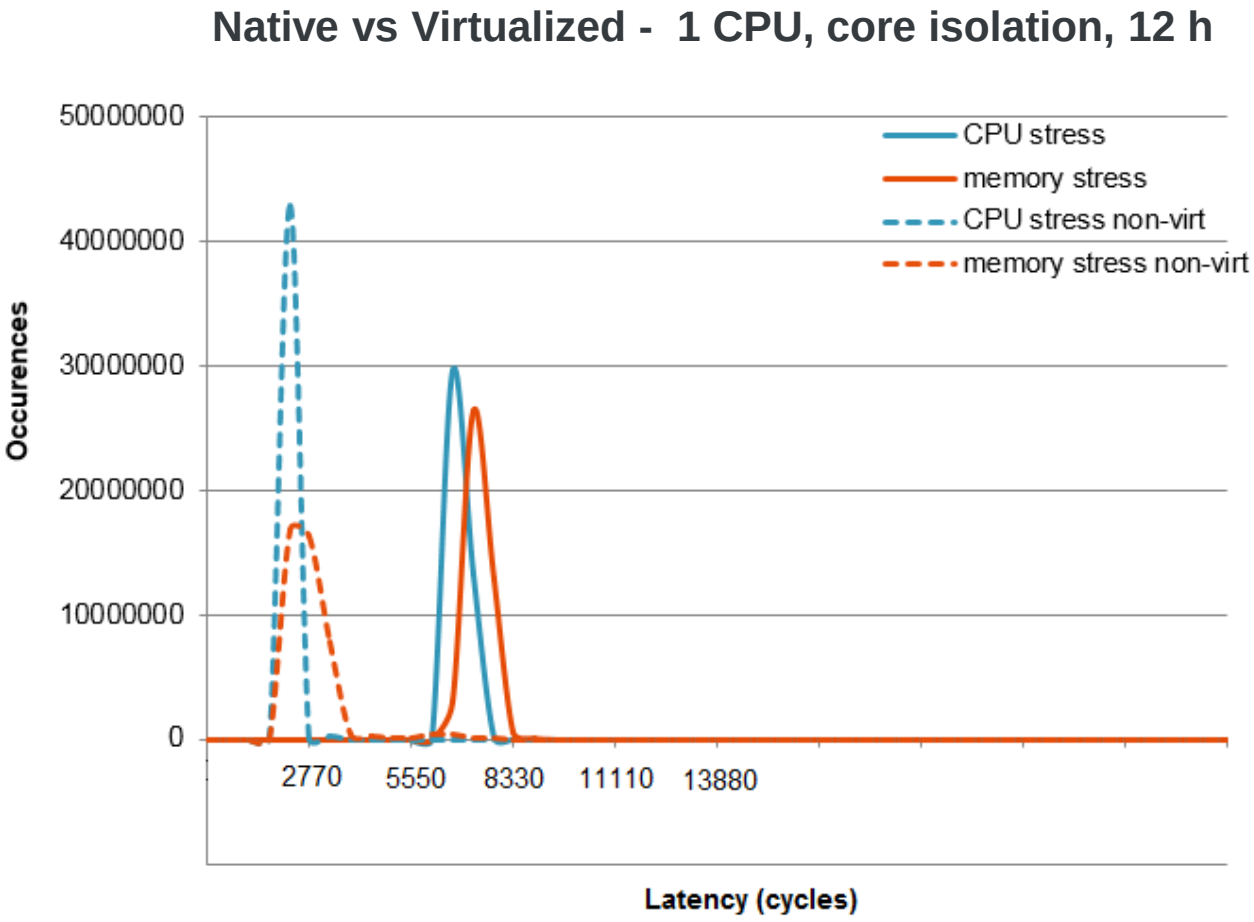
Networking Benchmarks

- Iperf

Perfomance tools

- Perf kvm
- CPU statistics
  - KVM ARM CPU accounting

RT KVM PA Latency Results - Cyclictest



CPU stress

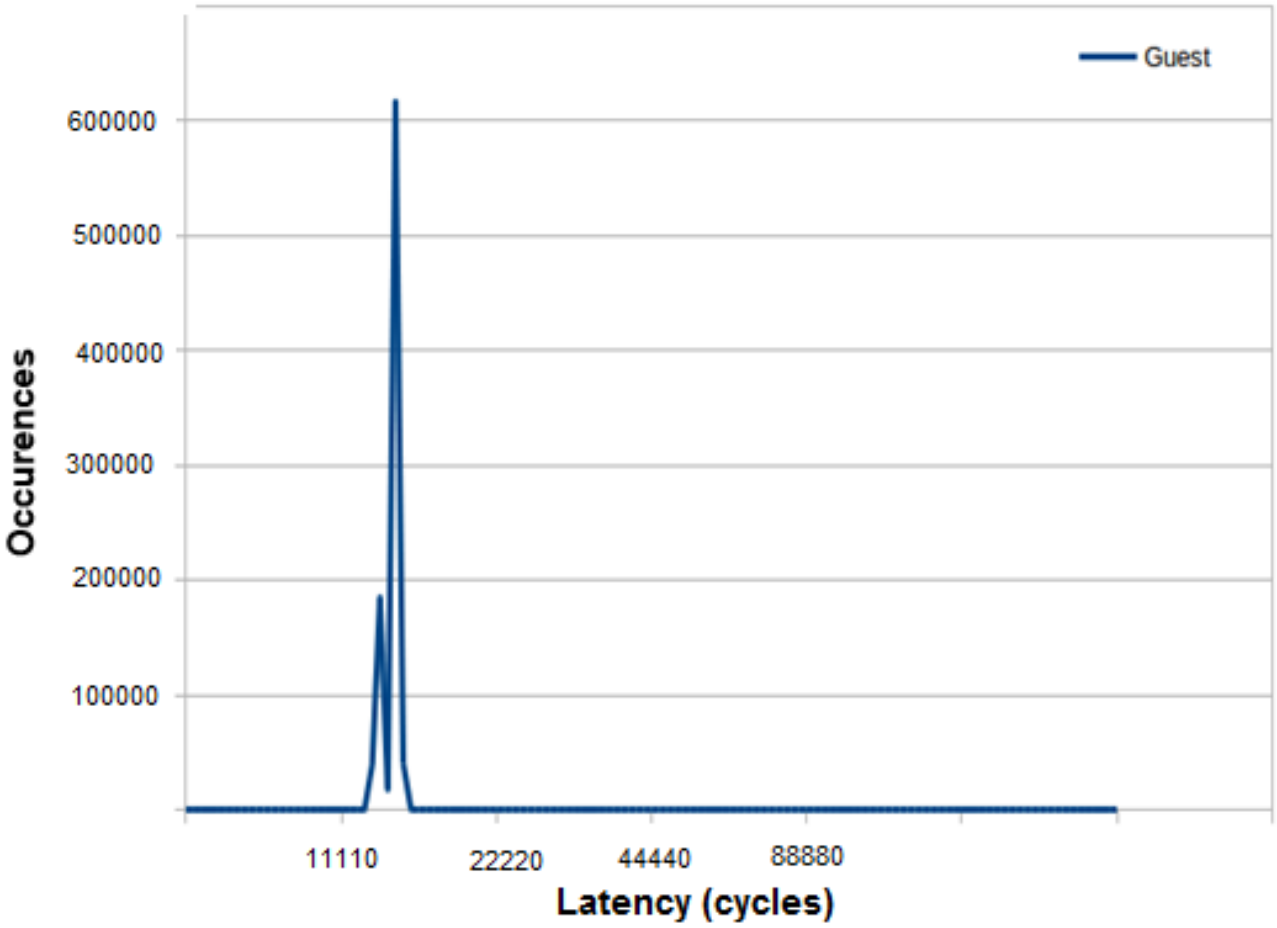
Latency (cycles)	Native	Virtual
Min	1660	2770
Avg	2220	6660
Max	**3330**	**11660**

Memory stress

Latency (cycles)	Native	Virtual
Min	1660	2220
Avg	2770	7220
Max	**13880**	**25550**

RT KVM PA Latency Results - Cyclictest

Virtualized pv-sched – host & guest coremark stress, 15 min



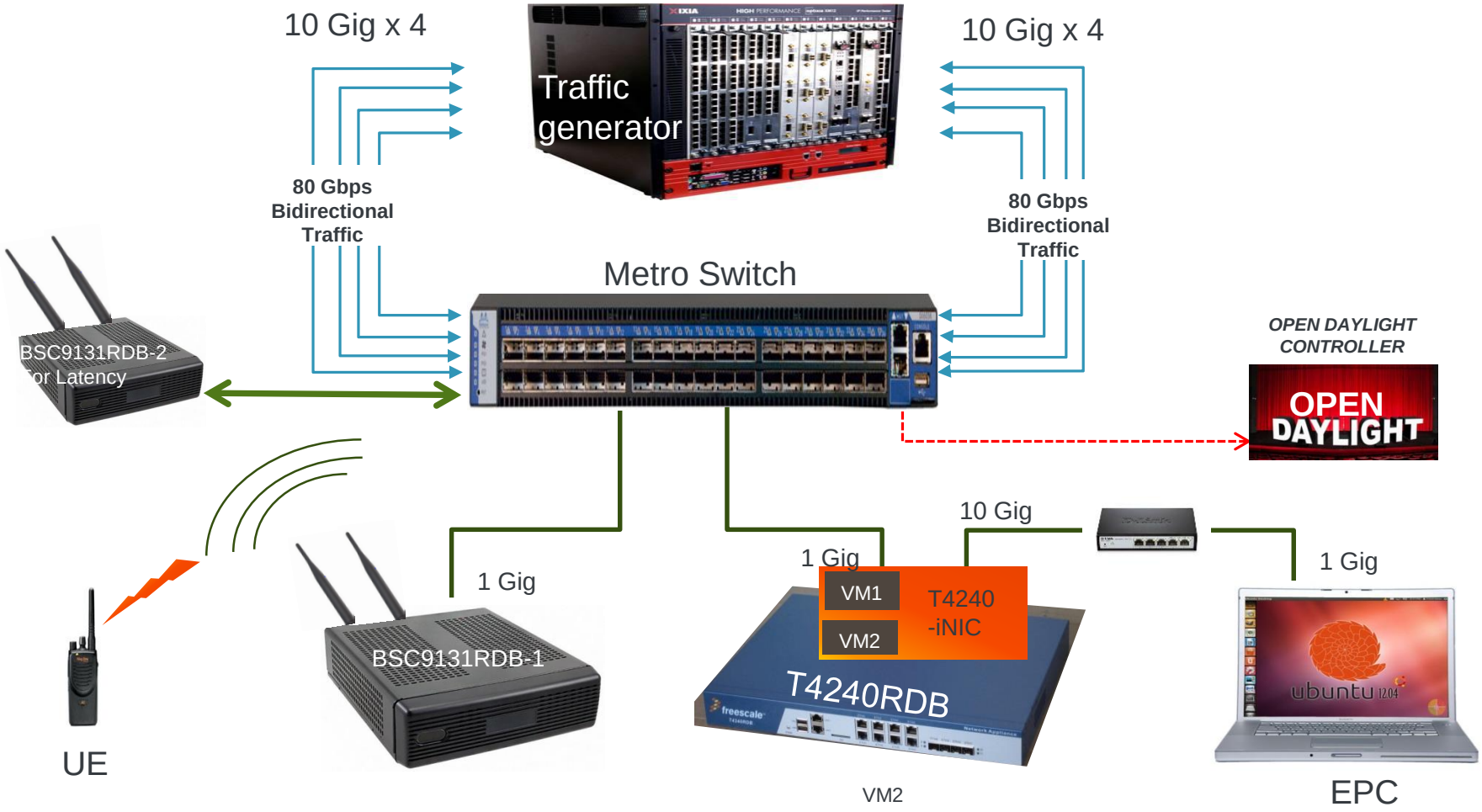
CPU stress (PV-SCH)

Latency (cycles)	Native
Min	5550
Avg	14440
Max	**27770**

Prio host coremark: 10

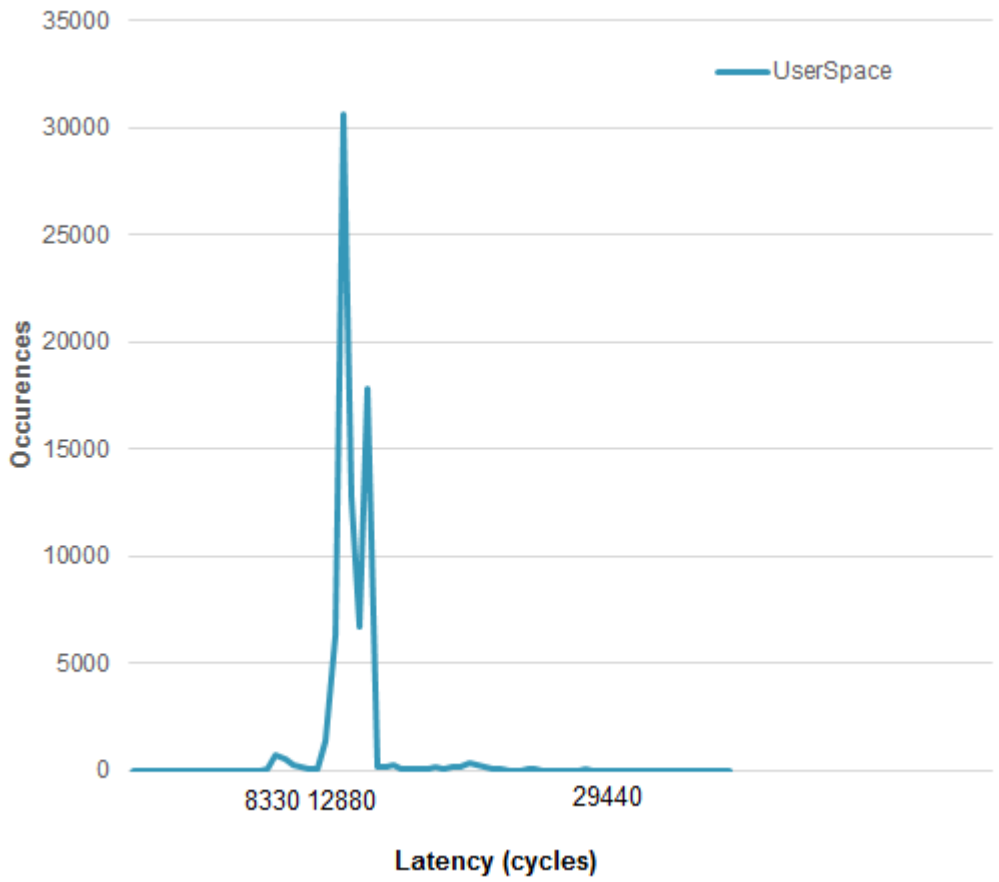
Prio guest coremark: 0

Virtual Base Station - Benchmarking Setup



RT KVM PA Latency Results - TTI External Interrupt

Virtualized – LTE L2 Stress



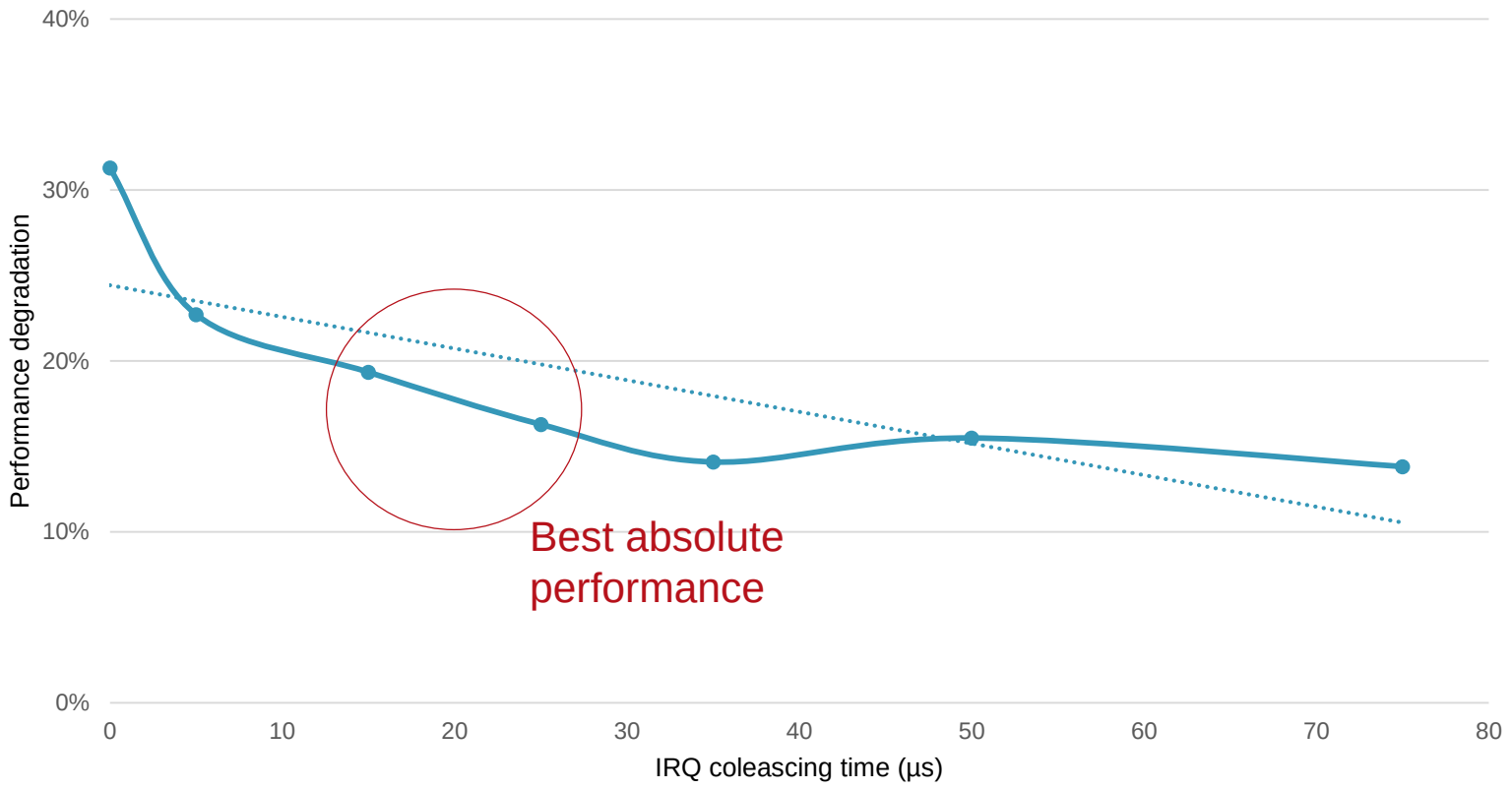
LTE L2 stress

Latency (us)	Virtual
Min	8330
Avg	13880
Max	**29444**

KVM ARMv8 - Networking performance degradation

Iperf TCP client 750 flows - DPAA 2 Direct Assignment

- 1 Network interface
- 1 VM, 1 vCPU



Future

- Scalability and performance
  - Stack disaggregation
  - Optimized I/O and accelerator access
  - CPU oversubscription
- Fast guest interrupt delivery
- IOMMU emulation

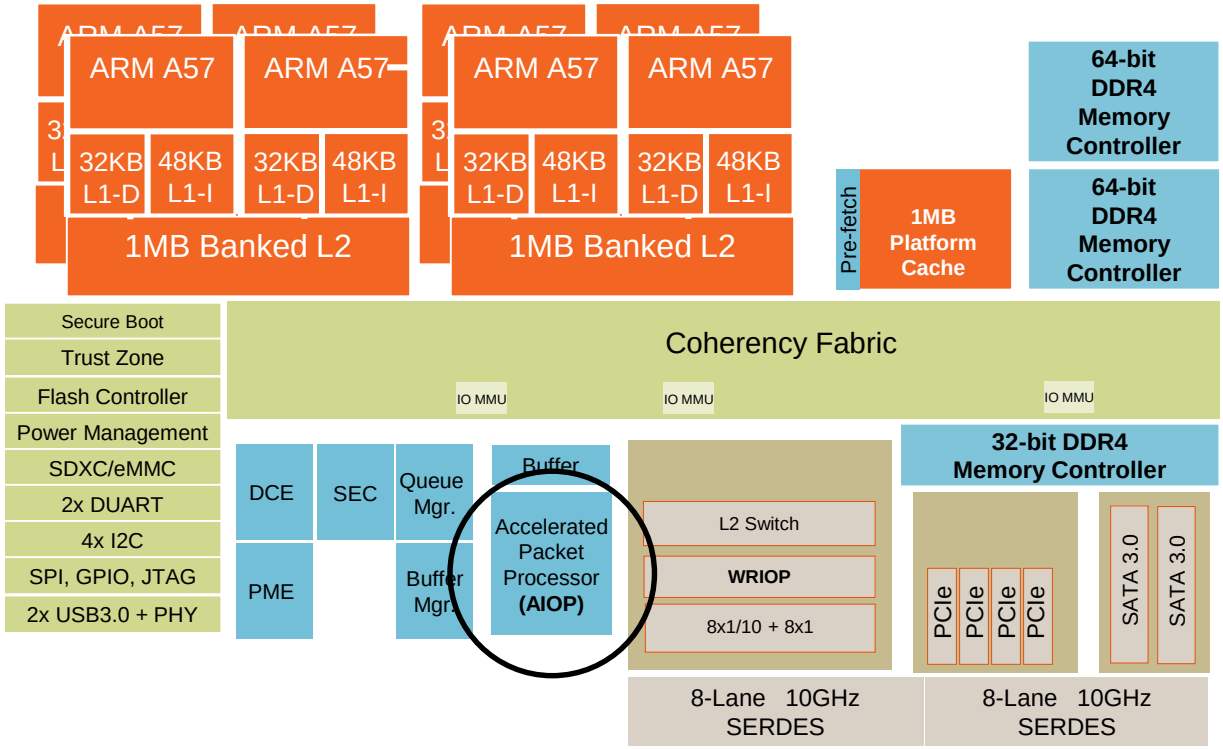
Summary

- “5G” networks are enabled by SDN and NFV
- Base station virtualization with RT KVM
- I/O virtualization using direct assignment and virtio
- NFV packet processing offload
- Virtual base station integration with OpenStack
- Low interrupt latency and low network performance degradation with KVM

Q & A

www.Freescale.com

LS2085A



General Purpose Processing

- 8x ARM A57 CPUs, 64b, 2.0GHz
  - 4MB Banked L2 cache
- HW L1 & L2 Prefetch Engines
- Neon SIMD in all CPUs
- 1MB L3 platform cache w/ECC
- 2x64b DDR4 up to 2.4GT/s

Accelerated I/O Processor

- 40Gbps Packet Processing
- 20Gbps SEC- crypto acceleration
- 15Gbps Pattern Match/RegEx
- 20Gbps Data Compression Engine
- 4MB Packet Express Buffer

Express Packet IO

- Supports1x8, 4x4, 4x2, 4x1 PCIe Gen3 controllers
- 2 x SATA 3.0, 2 x USB 3.0 with PHY

Network IO

- Wire Rate IO Processor:
  - 8x1/10GbE + 8x1G
  - XAUI/XFI/KR and SGMII
  - MACSec on up to 4x 1/10GbE

Other Parametrics

- 37.5x37.5 Flipchip
- 1mm Pitch
- 1292pins

Datapath Acceleration

- **SEC**- crypto acceleration
- **DCE** - Data Compression Engine
- **PME** – Pattern Matching Engine