Technical Reference Documentation
Getting Started

Rancher by SUSE

Intel® FlexRAN™ - SUSE Reference Solution Cloud-Native Setup

FlexRAN™ Deployment Guide on SUSE platform

SUSE Linux Enterprise Server 15.3 Real Time
SUSE Linux Enterprise Micro 5.2 Real Time
Rancher 2.6 by SUSE

Alex Zacharow, ISV Certification Engineer (SUSE)
Jose Betancourt, Director, Solution Partners & Alliances (SUSE)

Intel® FlexRAN™ - SUSE Reference Solution Cloud-Native Setup

FlexRAN™ Deployment Guide on SUSE platform

Date: 2023-03-02

Summary

This document provides the steps to build and configure Intel® FlexRAN™ 22.07 on SUSE Linux Enterprise 15 SP3 Real-Time (SLES 15 SP3 RT) and deploy a virtualized radio access network (vRAN) reference architecture based on FlexRAN™ containers on a Rancher Kubernetes Engine v2 (RKE2) cluster deployed on SUSE Linux Enterprise Micro 5.2 Real Time (SLE Micro RT) as a SUSE/Intel® reference design for telecommunications (telco).

Contents

1 Introduction

1.1 Motivation

Intel® FlexRAN™ is a reference implementation for cloud enabled wireless access virtual network functions (VNFs). It shows how to efficiently implement wireless access loads through flexible software architecture, Intel® Xeon® Scalable processors using Intel® Advanced Vector Extensions 512 (Intel® AVX 512) instruction set, and optimized network functions virtualization infrastructure (NFVi) with Intel-specific patches for the Data Plane Development Kit (DPDK).

With this *cloud-native setup*, you can simplify the installation and configuration of Intel® FlexRAN™, enabling you to focus on site-specific vRAN customizations vRAN.

1.2 Scope

In this guide, you learn to install and configure Intel's FlexRAN™ PHY Reference Design using SUSE Linux Enterprise Server Real Time as the base Operating System and SUSE Linux Enterprise Micro Real Time as target hosts for Rancher a RKE2 or K3s Kubernetes cluster with SUSE Rancher Server to manage this deployment.

1.3 Audience

This document is intended for developers and/or engineers in the telecommunications (telco) sector looking to build and test an Intel® FlexRAN™ test and/or proof-of-concept (PoC) environment with a SUSE stack combining real-time OS and Kubernetes orchestration and management.

2 Prerequisites

This section describes the hardware and software environment used to deploy Intel® FlexRAN™.

2.1 Access to Intel® FlexRAN™, Intel® oneAPI, and Intel® DPDK patches

Access to FlexRAN™ software and documentation is available to customers at Intel's Resource & Documentation Center (RDC) website. https://www.intel.com/content/www/us/en/documentation-resources/developer.html

If unable to access the link, please contact your Intel Field Application Engineer (FAR) for access accounts and credentials.

Intel® oneAPI as well as the Intel® patch for DPDK will be required to build FlexRAN™.

The platforms used for this guide had Intel® integrated GPU and as such required the installation of Intel® GPU drivers.

2.2 Hardware

This section lists the hardware configuration used as FlexRAN™ configuration and physical development nodes.

Component	Specification
Processor	Intel® Xeon® Silver 4316 @ 2.30Ghz
Memory	128 GB RAM
Network	Intel® vRAN ACC100-based accelerator Intel® E810 100Gb Ethernet controller
Storage	480GB SSD SATA Read Intensive 6Gbps 960GB Data Center NVMe

Note

For more details on server components, see Intel® FlexRAN™ reference documentation: *Installation Guide Software Release v22.07 (Doc. No.: 575834-15.0)* and *FlexRAN™ 5GNR Reference Solution 22.07 - PHY Software Documentation (Doc. No.: 603577)*.

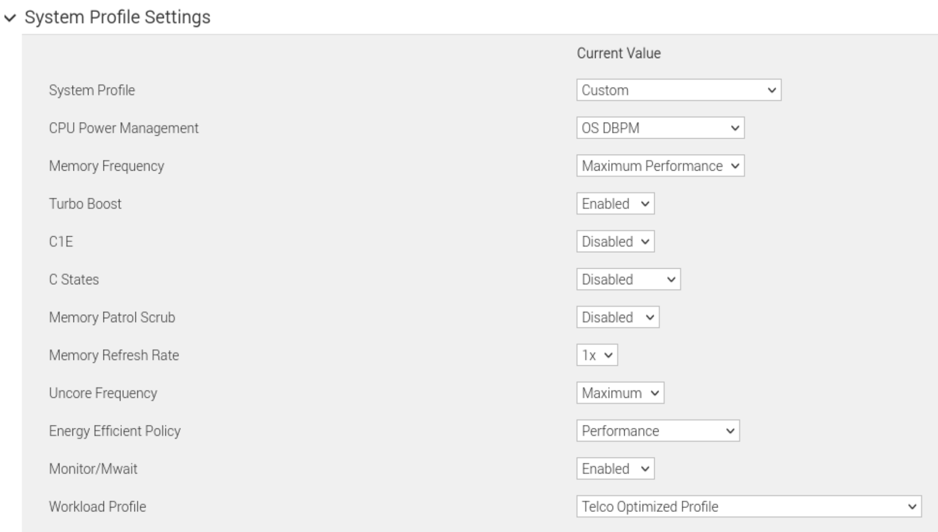
2.3 BIOS Configuration

A server's system BIOS provides runtime services for operating systems and performs hardware initialization during the booting process.

BIOS settings can influence how hardware behaves under different workloads.

Among the most important BIOS settings for implementing Intel® FlexRAN™ are the CPU p-states (optimization of the voltage and CPU fequency during operation) and c-states (optimization of the power consumption if a core does not have to execute any instructions).

BIOS configuration may be different for each server, but most modern servers should have similar settings.



For CPU power management, use a profile that allows for the OS to manipulate processor frequencies, such as *OS DBPM* or a similar control setting to allow the operating system to manipulate processor frequencies.

Other options such as a *Custom* profile with a *Telco Optimized* or *Maximum Performance* profile may be available, depending on the BIOS version.

Note

For more details, please review section 2.4.2 of *FlexRAN™ Software Reference Solution Cloud-Native Setup. (Intel® Doc. No. 575834-15.0)* and *BIOS Settings for FlexRAN™ Platforms Based on Intel® Xeon® Processors. (Doc. No.: 640685)*.

2.4 OS requirements

Intel® FlexRAN™ stipulates a real-time kernel, as listed in *Intel® FlexRAN™ Installation Guide (Doc. No.: 575834-15.0)*.

For the setup documented in this guide, we used a bare metal node as a development host running SUSE Linux Enterprise 15 SP3 Real-Time to preconfigure and test FlexRAN™ functionality and to build a container image before importing the container into a Rancher Kubernetes cluster (RKE2).

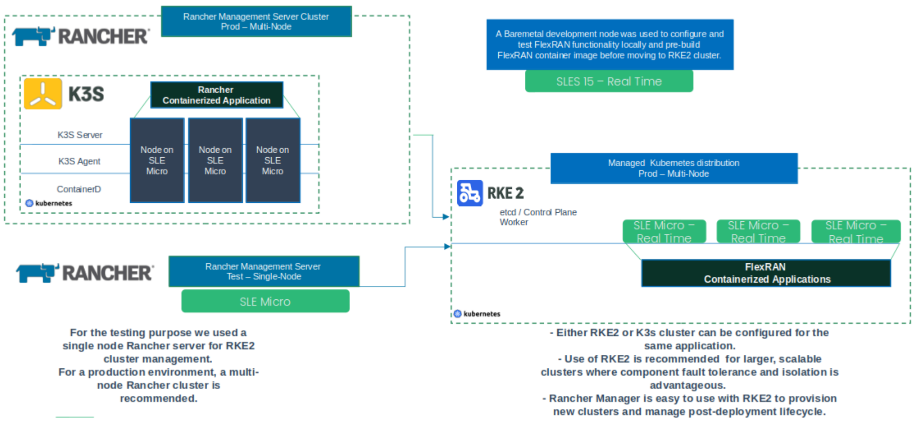


FIGURE 1: TEST SETUP DIAGRAM

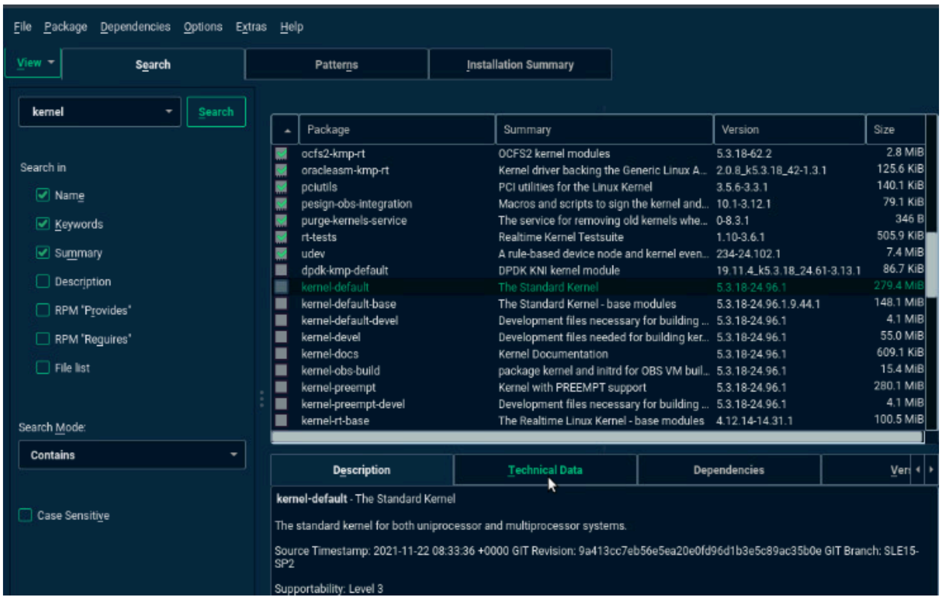
SUSE Linux Enterprise Real Time is a real time operating system designed to reduce latency and increase the predictability and reliability of time-sensitive, business-critical applications.

For more details about SLES RT please review https://www.suse.com/products/realtime/

2.4.1 Install SUSE Linux Enterprise Server 15 Real Time

Please refer to the SLES Setup Guide for step-by-step OS installation instructions: https://documentation.suse.com/sle-rt/15-SP3/

- When installing SLES 15 RT, during the installation make sure to unmark *kernel-default*.



- Verify *kernel-rt* is selected:

Note

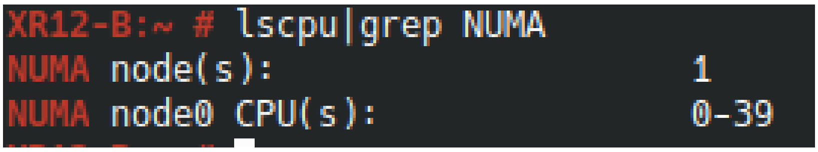
Add sufficient space to the /opt or /var directories since they will be used for most Intel® components and containers. We recommend 200 Gb of storage for these directories. Do not install FlexRAN™ under the root directory.

2.4.2 Real Time configuration

Isolate CPU cores with the following steps:

- Verify that *tuned* installed:

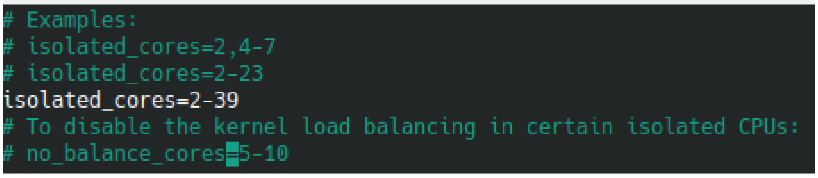
```
zypper in tuned*
```



In our installation setup, we have 1 socket and 40 cores.

- Add isolated cores to the configuration

```
vi /etc/tuned/cpu-partitioning-variables.conf
```

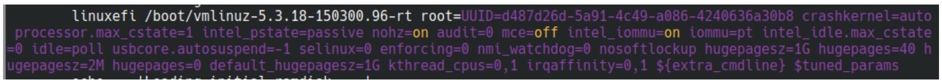


- Activate RT profile

```
tuned-adm profile cpu-partitioning
```

- For UEFI modify /boot/efi/EFI/sle_rt/grub.cfg as follows:

```
set tuned_params="skew_tick=1 nohz=on nohz_full=2-39 rcu_nocbs=2-39 nosoftlockup
 isolcpus=2-39"
```



Note

Settings are dependent on the number of CPUs and isolated cores. Please review section 2.4.3 of Intel's FlexRAN Cloud-Native Setup guide (document 575834-15.0).

- Save changes

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

- or for UEFI:

```
grub2-mkconfig -o /boot/efi/EFI/sle_rt/grub.cfg
```

- Reboot server and verify parameters:

```
grep tuned_params= /boot/grub2/grub.cfg
```

```
XR12-B:~ # grep tuned_params= /boot/grub2/grub.cfg
set tuned_params="skew_tick=1 nohz=on nohz_full=2-39 rcu_nocbs=2-39 nosoftlockup isolcpus=2-39"
```

```
cat /proc/cmdline
```

```
XR12-B:~ # cat /proc/cmdline
processor.max_cstate=1 intel_idle.max_cstate=1 skew_tick=1 hpc_cpusets BOOT_IMAGE=/boot/vmlinuz-5.3.18-150300.96-r
t root=UUID=d487d26d-5a91-4c49-a086-4240636a30b8 crashkernel=auto processor.max_cstate=1 intel_pstate=passive nohz
=on audit=0 mce=off intel_iommu=on iommu=pt intel_idle.max_cstate=0 idle=poll usbcore.autosuspend=-1 selinux=0 enf
orcing=0 nmi_watchdog=0 nosoftlockup hugepagesz=1G hugepages=40 hugepagesz=2M hugepages=0 default_hugepagesz=1G kt
hread_cpus=0,1 irqaffinity=0,1 skew_tick=1 nohz=on nohz_full=2-39 rcu_nocbs=2-39 nosoftlockup isolcpus=2-39
```

2.5 Set CPU Frequency

The AVX512 CPU frequency of your specific CPU should be adjusted according to Figure 4 of Intel's doc Reference Number: 637779, Revision: 1.2 3rd Gen Intel® Xeon® Scalable Processors, Codename Ice Lake NDA Specification Update June 2021 or #613537 for Skylake processor family

					# of active cores / maximum core frequency in turbo mode (GHz)																									
SKU	Cores	LLC (MB)	TDP (W)	Base AVX 512 Core Freq (GHz)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
5320	26	39	185	1.6	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.1	3.1	2.9	2.9	2.7	2.7	2.7	2.7	2.6	2.6	2.6	2.6
6342	24	36	230	2.1	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.2	3.2		
6338T	24	36	165	1.5	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.1	3.1	2.9	2.9	2.6	2.6	2.6	2.6	2.5	2.5	2.5	2.5		
6336Y	24	36	185	1.7	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.2	3.2	3.2	3.2	3.1	3.1	2.8	2.8	2.8	2.8	2.8	2.8		
6312U	24	36	185	1.8	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.3	3.3	3.2	3.2	3.0	3.0	3.0	3.0	2.8	2.8	2.8	2.8		
5318Y	24	36	165	1.5	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.1	3.1	2.9	2.9	2.6	2.6	2.6	2.6	2.5	2.5	2.5	2.5		
5318S	24	36	165	1.5	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.1	3.1	2.9	2.9	2.6	2.6	2.6	2.6	2.5	2.5	2.5	2.5		
5318N	24	36	150	1.5	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.0	3.0	2.7	2.7	2.7	2.7	2.6	2.6	2.5	2.5	2.5	2.5		
5320T	20	30	150	1.6	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.2	3.2	3.0	3.0	2.9	2.9	2.8	2.8	2.7	2.7						
4316	20	30	150	1.6	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.1	3.1	2.9	2.9	2.8	2.8	2.7	2.7	2.6	2.6						
6326	16	24	185	2.1	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.2	3.2	3.2	3.2	3.2	3.2	3.1	3.1										
4314	16	24	135	1.7	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.1	3.1	3.0	3.0	2.8	2.8	2.7	2.7										
5317	12	18	150	2.3	3.4	3.4	3.4	3.4	3.4	3.4	3.3	3.3	3.3	3.3	3.2	3.2														

For this configuration setup, an Intel® Xeon® 4316 processor with 2.6 GHz was used.

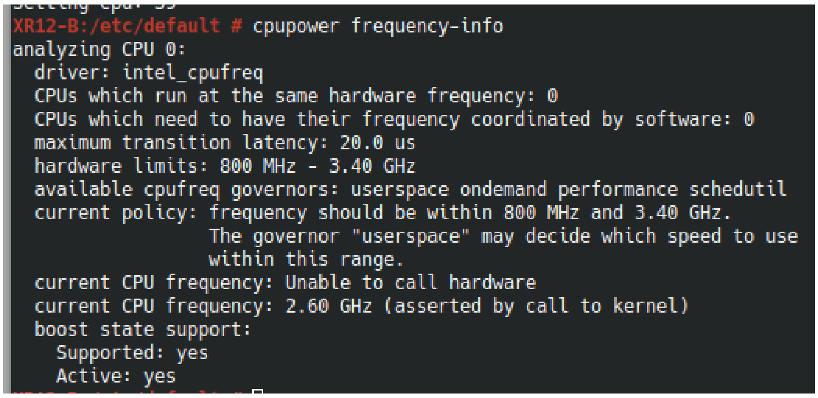
There are two options to setup your CPU frequency:

- **Use cpupower tool**

By running

```
cpupower frequency-info
```

you can check available frequencies for your CPU and drivers.



```
XR12-B:/etc/default # cpupower frequency-info
analyzing CPU 0:
  driver: intel_cpufreq
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency: 20.0 us
  hardware limits: 800 MHz - 3.40 GHz
  available cpufreq governors: userspace ondemand performance schedutil
  current policy: frequency should be within 800 MHz and 3.40 GHz.
                  The governor "userspace" may decide which speed to use
                  within this range.
  current CPU frequency: Unable to call hardware
  current CPU frequency: 2.60 GHz (asserted by call to kernel)
  boost state support:
    Supported: yes
    Active: yes
```

In this configuration example the intel_cpufreq driver was used. The userspace governor is available with the older acpi-cpufreq driver (which would be automatically used if you disable intel_pstate at boot time; you then set the governor/frequency with cpupower)

- Set intel_pstate driver to passive in grub (intel_pstate=passive):

```
echo passive | sudo tee /sys/devices/system/cpu/intel_pstate/status
```

or add intel_pstate=passive to the grub:

```
modprobe cpufreq_userspace
```

- Set cpu governor to userspace:

```
cpupower frequency-set --governor userspace
```

- Set frequency according to the AVX-512 table (2600MHz in this case):

```
cpupower --cpu all frequency-set --freq 2600MHz
```

Note

It is important to set C-state and P-state on the Bios settings as well as on the kernel side. If you don't do this, you won't be able to change governors from the cpupower command and set the cpu frequency. Also, make sure that the BIOS can be changed from the OS by proper setting.

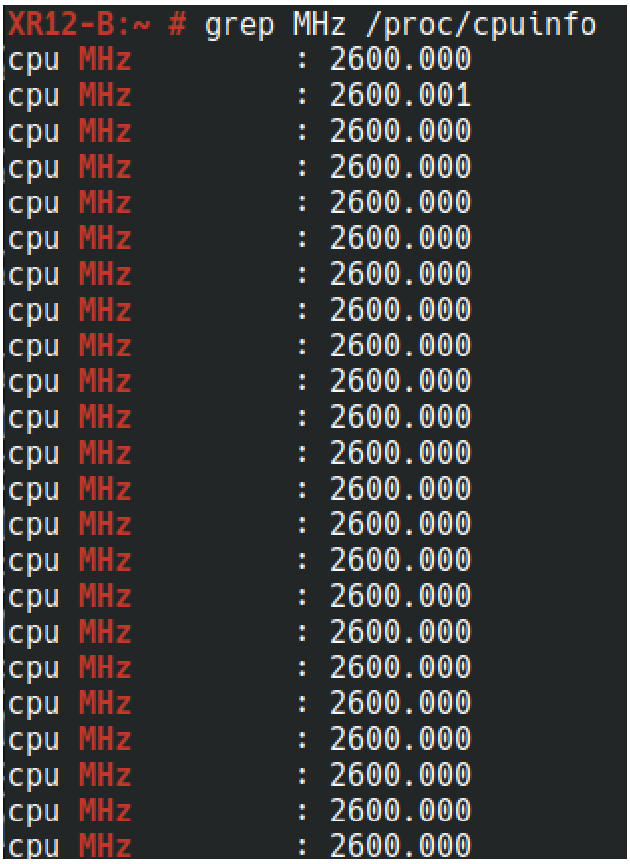
- Verify that settings applied by running:

```
turbostat -i 1
```

```
^CCore  CPU     Avg_MHz Busy%   Bzy_MHz TSC_MHz IRQ     SMI     CPU%c1  CPU%c6  CoreTmp PkgTmp  Pkg%pc2 Pkg%pc6 PkgWatt RAMWatt PKG_%   RAM_%
-       -       2594    100.00  2600    1596    6661    0       0.00    0.00    45      45      0.00    0.00    107.91  16.64   0.00    0.00
0       0       2594    100.00  2600    1596    166     0       0.00    0.00    44      45      0.00    0.00    107.91  16.64   0.00    0.00
0       20      2594    100.00  2600    1596    176     0       0.00
1       2       2594    100.00  2600    1596    166     0       0.00    0.00    44
1       22      2594    100.00  2600    1596    165     0       0.00
2       4       2594    100.00  2600    1596    166     0       0.00    0.00    44
2       24      2594    100.00  2600    1596    165     0       0.00
3       6       2594    100.00  2600    1596    166     0       0.00    0.00    43
3       26      2594    100.00  2600    1596    166     0       0.00
4       8       2594    100.00  2600    1596    166     0       0.00    0.00    43
4       28      2594    100.00  2600    1596    166     0       0.00
5       10      2594    100.00  2600    1596    166     0       0.00    0.00    43
5       30      2594    100.00  2600    1596    166     0       0.00
6       12      2594    100.00  2600    1596    166     0       0.00    0.00    42
6       32      2594    100.00  2600    1596    166     0       0.00
```

- You can also check with other available tools:

```
XR12-B:~ # cpupower monitor -m 'Mperf'
    | Mperf
 CPU| C0    | Cx    | Freq
   0| 99.94|  0.06|  2593
  20| 99.94|  0.06|  2593
   2| 99.94|  0.06|  2593
  22| 99.94|  0.06|  2593
   4| 99.94|  0.06|  2593
  24| 99.94|  0.06|  2593
   6| 99.94|  0.06|  2593
  26| 99.94|  0.06|  2593
   8| 99.94|  0.06|  2593
  28| 99.94|  0.06|  2593
  10| 99.94|  0.06|  2593
  30| 99.94|  0.06|  2593
  12| 99.94|  0.06|  2593
  32| 99.94|  0.06|  2593
  14| 99.94|  0.06|  2593
  34| 99.94|  0.06|  2593
  16| 99.94|  0.06|  2593
  36| 99.94|  0.06|  2593
  18| 99.94|  0.06|  2593
  38| 99.94|  0.06|  2593
   1| 99.94|  0.06|  2593
```



```
XR12-B:~ # grep MHz /proc/cpuinfo
cpu MHz         : 2600.000
cpu MHz         : 2600.001
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
```

- **The second option to change AVX512 frequency is to install Intel® msr-tools with the following commands:**

```
git clone https://github.com/intel/msr-tools/
cd msr-tools/
git checkout msr-tools-1.3
make
modprobe msr
```

- Create bash script setFreq.sh with the following context:

```
#!/bin/bash

cpupower frequency-set -g performance

for i in {0..39}

do

/home/Intel/msr-tools/msr-tools/wrmsr -p $i 0x199 0x1A00
```

```
done

#Set Uncore max frequency

/home/Intel/msr-tools/msr-tools/wrmsr -p 0 0x606A6 0x1A00

/home/Intel/msr-tools/msr-tools/wrmsr -p 39 0x606A6 0x1A00
```

Note

Values in the script were taking from Intel® document #637779 (for Ice Lake family) specific to your CPU avx512 numbers. (2.6 GHz in the above example).

Table 2. 3rd Gen Intel® Xeon® Scalable Processors Identification

Processor Number	QDF/S-Spec Number	Die	Stepping	CPUID	Speed (GHz)	DDR4 (MHz)	TDP (W)	# of Cores	LLC Cache Size (MB)	Max. Supported Sockets/ Intel UPI Links
4309Y	SRKXS	HCC	M1	0x606A6	2.8	2667	105	8	12	2/2
6342	QXRU	HCC	M1	0x606A6	2.8	3200	230	24	36	2/3
6338T	QXS3	HCC	M1	0x606A6	2.1	3200	165	24	36	2/3
6336Y	QXRV	HCC	M1	0x606A6	2.4	3200	185	24	36	2/3
6334	QXRQ	HCC	M1	0x606A6	3.6	3200	165	8	12	2/3
6326	QXS7	HCC	M1	0x606A6	2.9	3200	185	16	24	2/3
6312U	QXRW	HCC	M1	0x606A6	2.4	3200	185	24	36	1/0
5320T	QXS6	HCC	M1	0x606A6	2.3	2933	150	20	30	2/3
5320	QXRT	HCC	M1	0x606A6	2.2	2933	185	26	39	2/3
5318Y	QXS2	HCC	M1	0x606A6	2.1	2933	165	24	36	2/3
5318S	QXRX	HCC	M1	0x606A6	2.1	2933	165	24	36	2/3
5318N	QXS4	HCC	M1	0x606A6	2.1	2667	150	24	36	2/2
5317	QXRM	HCC	M1	0x606A6	3.0	2933	150	12	18	2/3
5315Y	QXRR	HCC	M1	0x606A6	3.2	2933	140	8	12	2/3
4316	QXS5	HCC	M1	0x606A6	2.3	2667	150	20	30	2/2
4314	QXS8	HCC	M1	0x606A6	2.4	2667	135	16	24	2/2
4310T	QXRP	HCC	M1	0x606A6	2.3	2667	105	10	15	2/2
4310	QXRN	HCC	M1	0x606A6	2.1	2667	120	12	18	2/2
4309Y	QXRS	HCC	M1	0x606A6	2.8	2667	105	8	12	2/2

Run the above bash script with your specific numbers which should be changed to the required frequency and verify that required frequency was applied.

- Review performance with a *cyclictest*:

```
XR12-B:~ # taskset -c 0-19 cyclictest -m -p95 -h 15 -a 1-19 -t 19 --mainaffinity=0
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 2.76 3.02 2.86 2/1182 14957

T: 0 ( 8667) P:95 I:1000 C:1726624 Min:      1 Act:    2 Avg:    2 Max:      17
T: 1 ( 8668) P:95 I:1000 C:1726630 Min:      2 Act:    2 Avg:    2 Max:      15
T: 2 ( 8669) P:95 I:1000 C:1726629 Min:      2 Act:    3 Avg:    2 Max:      12
T: 3 ( 8670) P:95 I:1000 C:1726629 Min:      2 Act:    2 Avg:    2 Max:      15
T: 4 ( 8671) P:95 I:1000 C:1726629 Min:      2 Act:    2 Avg:    2 Max:      13
T: 5 ( 8672) P:95 I:1000 C:1726628 Min:      2 Act:    2 Avg:    2 Max:      13
T: 6 ( 8673) P:95 I:1000 C:1726628 Min:      2 Act:    2 Avg:    2 Max:      11
T: 7 ( 8674) P:95 I:1000 C:1726628 Min:      2 Act:    2 Avg:    2 Max:      11
T: 8 ( 8675) P:95 I:1000 C:1726627 Min:      2 Act:    2 Avg:    2 Max:      11
T: 9 ( 8676) P:95 I:1000 C:1726627 Min:      2 Act:    2 Avg:    2 Max:      10
T:10 ( 8677) P:95 I:1000 C:1726627 Min:      2 Act:    2 Avg:    2 Max:      13
T:11 ( 8678) P:95 I:1000 C:1726627 Min:      2 Act:    2 Avg:    2 Max:      15
T:12 ( 8679) P:95 I:1000 C:1726626 Min:      2 Act:    2 Avg:    2 Max:      13
T:13 ( 8680) P:95 I:1000 C:1726626 Min:      2 Act:    2 Avg:    2 Max:      11
T:14 ( 8681) P:95 I:1000 C:1726625 Min:      2 Act:    2 Avg:    2 Max:      10
T:15 ( 8682) P:95 I:1000 C:1726625 Min:      2 Act:    2 Avg:    2 Max:       8
T:16 ( 8683) P:95 I:1000 C:1726625 Min:      2 Act:    2 Avg:    2 Max:      12
T:17 ( 8684) P:95 I:1000 C:1726625 Min:      2 Act:    3 Avg:    2 Max:      10
T:18 ( 8685) P:95 I:1000 C:1726624 Min:      2 Act:    3 Avg:    2 Max:      11
```

For more details, please review [SLE RT Hardware Testing] https://documentation.suse.com/sle-rt/15-SP3/pdf/article-hardware-testing_color_en.pdf ↗

2.6 Install Intel® oneAPI

Install Intel® GPU drivers since our platform has an Intel® GPU.

The installation of the drivers also eliminate potential prerequisite failure for oneAPI.

Review https://dgpu-docs.intel.com/installation-guides/suse/suse-15sp3.html ↗ for more details.

```
zypper addrepo -r https://repositories.intel.com/graphics/sles/15sp3/intel-graphics.repo

zypper install   intel-opencl   intel-media-driver libmfx1   intel-level-zero-gpu level-
zero
```

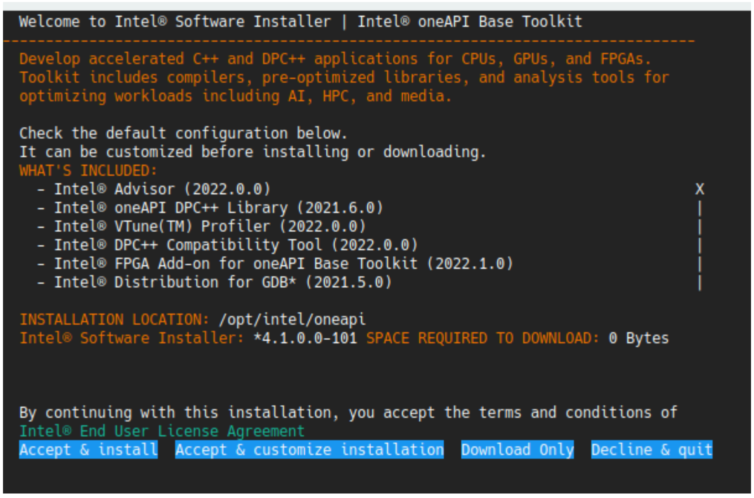
Download and install Intel® oneAPI

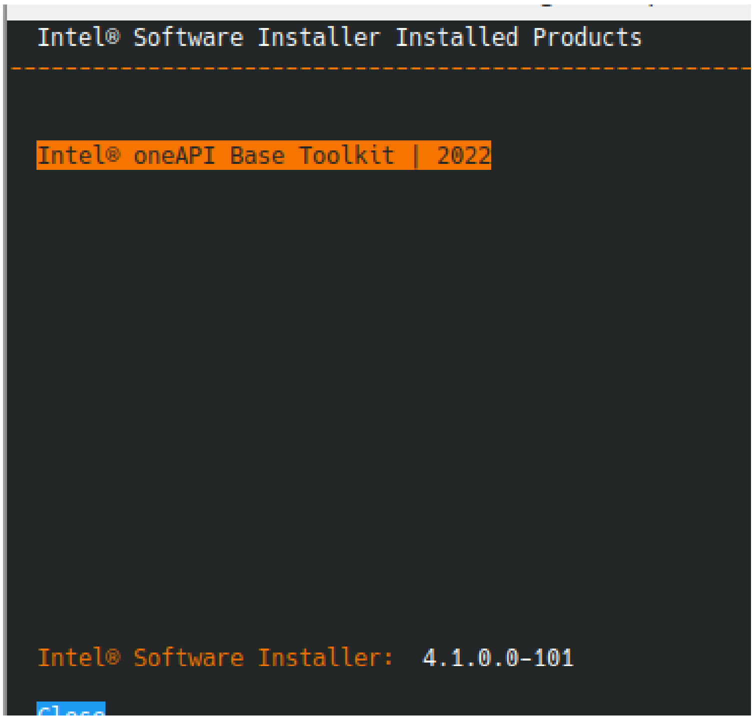
```
wget https://registrationcenter-download.intel.com/akdlm/irc_nas/18236/
l_BaseKit_p_2021.4.0.3422_offline.sh

bash l_BaseKit_p_2021.4.0.3422_offline.sh
```

Note

Make sure that the installation directory has enough space. Intel® oneAPI utilizes approximately 40Gb of space.


Welcome to Intel® Software Installer | Intel® oneAPI Base Toolkit
Develop accelerated C++ and DPC++ applications for CPUs, GPUs, and FPGAs.
Toolkit includes compilers, pre-optimized libraries, and analysis tools for
optimizing workloads including AI, HPC, and media.
Check the default configuration below.
It can be customized before installing or downloading.
WHAT'S INCLUDED:
- Intel® Advisor (2022.0.0)
- Intel® oneAPI DPC++ Library (2021.6.0)
- Intel® VTune(TM) Profiler (2022.0.0)
- Intel® DPC++ Compatibility Tool (2022.0.0)
- Intel® FPGA Add-on for oneAPI Base Toolkit (2022.1.0)
- Intel® Distribution for GDB* (2021.5.0)
INSTALLATION LOCATION: /opt/intel/oneapi
Intel® Software Installer: *4.1.0.0-101 SPACE REQUIRED TO DOWNLOAD: 0 Bytes
By continuing with this installation, you accept the terms and conditions of
Intel® End User License Agreement
Accept & install
Accept & customize installation
Download Only
Decline & quit


Intel® Software Installer Installed Products
Intel® oneAPI Base Toolkit | 2022
Intel® Software Installer: 4.1.0.0-101

- Source the environment and verify installed version:

```
XR12-B:/opt/intel/oneapi # source /opt/intel/oneapi/setvars.sh

:: initializing oneAPI environment ...
   -bash: BASH_VERSION = 4.4.23(1)-release
   args: Using "$@" for setvars.sh arguments:
:: advisor -- latest
:: ccl -- latest
:: compiler -- latest
:: dal -- latest
:: debugger -- latest
:: dev-utilities -- latest
:: dnnl -- latest
:: dpcpp-ct -- latest
:: dpl -- latest
:: intelpython -- latest
:: ipp -- latest
:: ippcp -- latest
:: mkl -- latest
:: mpi -- latest
:: tbb -- latest
:: vpl -- latest
:: vtune -- latest
:: oneAPI environment initialized ::

XR12-B:/opt/intel/oneapi # icx -v
Intel(R) oneAPI DPC++/C++ Compiler 2022.0.0 (2022.0.0.20211123)
Target: x86_64-unknown-linux-gnu
Thread model: posix
InstalledDir: /opt/intel/oneapi/compiler/2022.0.2/linux/bin-llvm
Found candidate GCC installation: /usr/lib64/gcc/x86_64-suse-linux/7
Selected GCC installation: /usr/lib64/gcc/x86_64-suse-linux/7
Candidate multilib: .;@m64
Selected multilib: .;@m64
```

- Make sure that GCC is installed to work with ICX compiler:

```
XR12-B:/etc/default # gcc --version
gcc (SUSE Linux) 7.5.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

3 Intel® FlexRAN™ Installation

For complete installation details, please review the compilation tools section of *FlexRAN 5GNR Reference Solution 22.07 PHY Software Documentation* - Document #603577

Make sure that your instance has installed *cmake, meson, and ninja.*

In order to build the L1 application and L1 standalone Test Application, the following steps are required (in order):

3.1 Install pkgconf tool

```
zypper in automake
zypper in libtool
git clone https://github.com/pkgconf/pkgconf.git
cd pkgconf/
./configure
make
make install
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
```

3.2 Download and Install DPDK

Note

Don't use /root directory for the installation. Intel's DPDK patch is required.

```
wget http://static.dpdk.org/rel/dpdk-21.11.tar.xz
tar xf dpdk-21.11.tar.xz
export RTE_SDK=/var/dpdk/dpdk-21.11
```

- Copy patch to RTE_SDK directory and apply dpdk patch:

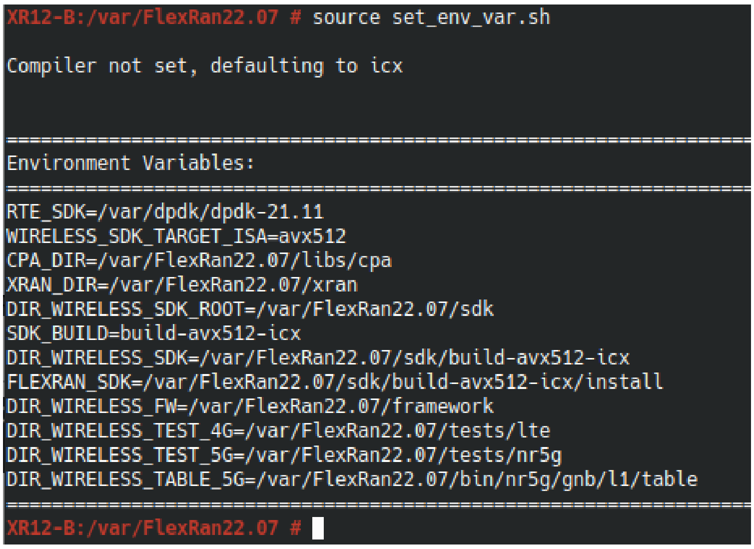
```
patch -p1 < dpdk_patch_21.11.patch
```

3.3 Download and install FlexRAN™

Download FlexRAN™ release pursuant to *Intel® FlexRAN™ Software Reference Solution Release Announcement* - Document 645964.

- Extract file and source the environment:

```
tar -zxvf FlexRan-22.07.tar.gz
./extract.sh
export RTE_SDK=/var/dpdk/dpdk-21.11
source ./set_env_var.sh
```



3.4 Compile SDK

- Get *gcc11-c++*:

```
zypper in gcc11-c++
```

- Export PKG_CONFIG_PATH:

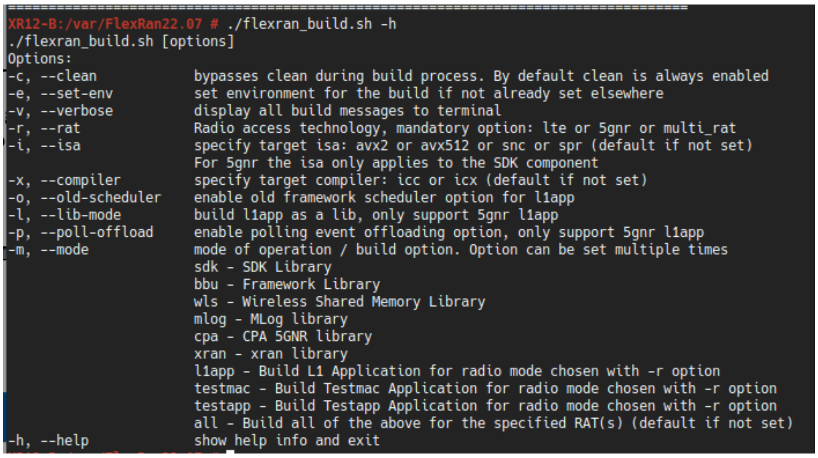
```
export PKG_CONFIG_PATH=$DIR_WIRELESS_SDK/pkgcfg:$PKG_CONFIG_PATH
```

- Source oneAPI:

```
source /opt/intel/oneapi/setvars.sh --force

export PATH=/opt/intel/oneapi/compiler/2022.0.2/linux/bin-llvm/:$PATH
```

- Review possible compilation options from *./flexran_build.sh -h* command:



```
XR12-B:/var/FlexRan22.07 # ./flexran_build.sh -h
./flexran_build.sh [options]
Options:
-c, --clean            bypasses clean during build process. By default clean is always enabled
-e, --set-env          set environment for the build if not already set elsewhere
-v, --verbose          display all build messages to terminal
-r, --rat              Radio access technology, mandatory option: lte or 5gnr or multi_rat
-i, --isa              specify target isa: avx2 or avx512 or snc or spr (default if not set)
                       For 5gnr the isa only applies to the SDK component
-x, --compiler         specify target compiler: icc or icx (default if not set)
-o, --old-scheduler    enable old framework scheduler option for l1app
-l, --lib-mode         build l1app as a lib, only support 5gnr l1app
-p, --poll-offload     enable polling event offloading option, only support 5gnr l1app
-m, --mode             mode of operation / build option. Option can be set multiple times
                       sdk - SDK Library
                       bbu - Framework Library
                       wls - Wireless Shared Memory Library
                       mlog - MLog library
                       cpa - CPA 5GNR library
                       xran - xran library
                       l1app - Build L1 Application for radio mode chosen with -r option
                       testmac - Build Testmac Application for radio mode chosen with -r option
                       testapp - Build Testapp Application for radio mode chosen with -r option
                       all - Build all of the above for the specified RAT(s) (default if not set)
-h, --help             show help info and exit
```

- Compile SDK:

```
./flexran_build.sh -x icx -e -r 5gnr -i avx512 -m sdk
```

Note

The FlexRAN™ SDK libraries must be built first to the provided path before starting the DPDK build process so that software FEC libraries are present.

3.5 Patch and Compile DPDK

```
zypper in python3-pyelftools.rpm
```

- Create dpdk script:

```
vi dpdk-dep.sh
```

```
#! /bin/bash
work_path=$PWD
sdk_path= /var/FlexRan22.07/sdk
echo "------------build base dpdk ------------------"
cd $RTE_SDK; meson build; cd build; meson configure -Dflexran_sdk=$sdk_path/build-avx512-
icx/install; ninja
```

- Run dpdk script:

```
./dpdk-dep.sh
```

- Create dpdk-kmods:

```
git clone http://dpdk.org/git/dpdk-kmods
cd dpdk-kmods/linux/igb_uio/
make
modprobe uio
insmod $RTE_SDK_KMOD/linux/igb_uio/igb_uio.ko
export RTE_SDK_KMOD=/var/dpdk/dpdk-kmods
```

3.6 Build the L1 Application, L1 Standalone Test Application, and Test MAC in Linux:

Verify that you have *numa*, libhuge*, and libnuma-dev** installed.

- Mount hugepages:

```
mount -t hugetlbfs nodev /mnt/huge
```

- Compile for 5G New Radio (5gnr) solution:

```
./flexran_build.sh -x icx -e -r 5gnr
```

- Compile for Long Term Evolution (LTE) solution:

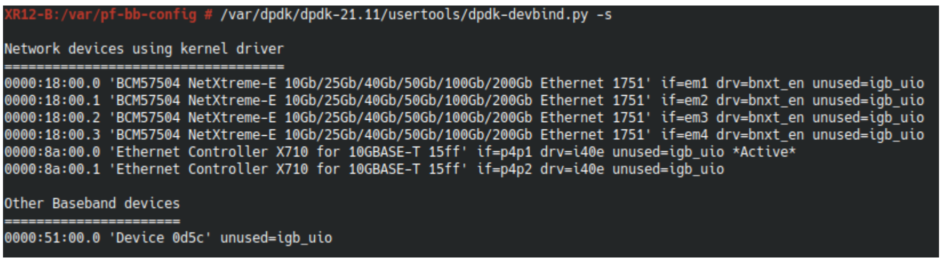
```
./flexran_build.sh -x icx -e -r lte -i avx512
```

After following above steps, upon a successful build, a new L1 application file <install_dir>/bin/nr5g/gnb/l1 will be created. L1 standalone Test Application will be created in <install_dir>/tests/nr5g/nr5g_testapp

For ACC100 acceleration:

- Verify acc card:

```
lspci | grep acc
51:00.0 Processing accelerators: Intel Corporation Device 0d5c
```



When using Mount Bryce (ACC100) dedicated accelerator card, follow the instructions at https://github.com/intel/pf-bb-config

```
git clone https://github.com/intel/pf-bb-config
cd pf-bb-config/
make
```

`For Physical Function (PF) option:`

- Bind the PF with the igb_uio module (or alternatively with pci-pf-stub):

```
/var/dpdk/dpdk-21.11/usertools/dpdk-devbind.py --bind=igb_uio 51:00.0
```

- Configure the devices using the pf_bb_config application:

```
/var/dpdk/dpdk-21.11/usertools/dpdk-devbind.py --bind=igb_uio 52:00.0 52:00.1
```

```
XR12-B:/var/pf-bb-config # ./pf_bb_config ACC100 -c acc100/acc100_config_2vf_4g5g.cfg
== pf_bb_config Version #VERSION_STRING# ==
Queue Groups: 2 5GUL, 2 5GDL, 2 4GUL, 2 4GDL
Number of 5GUL engines 8
Configuration in VF mode
 ROM version MM 99AD92
DDR Training completed in 1369 msPF ACC100 configuration complete
ACC100 PF [0000:51:00.0] configuration complete!
```

`For Virtual Function (VF) option:`

- Create 2 VFs from the PF:

```
============================
XR12-B:/var/pf-bb-config/acc100 # echo 2 | sudo tee /sys/bus/pci/devices/0000:51:00.0/max_vfs
2
```

- Check available interfaces:

```
/opt/dpdk/dpdk-stable-20.11.3/usertools/dpdk-devbind.py -s
```

```
XR12-B:/var/pf-bb-config/acc100 # /var/dpdk/dpdk-21.11/usertools/dpdk-devbind.py -s

Network devices using kernel driver
===================================
0000:18:00.0 'BCM57504 NetXtreme-E 10Gb/25Gb/40Gb/50Gb/100Gb/200Gb Ethernet 1751' if=em1 drv=bnxt_en unused=igb_uio
0000:18:00.1 'BCM57504 NetXtreme-E 10Gb/25Gb/40Gb/50Gb/100Gb/200Gb Ethernet 1751' if=em2 drv=bnxt_en unused=igb_uio
0000:18:00.2 'BCM57504 NetXtreme-E 10Gb/25Gb/40Gb/50Gb/100Gb/200Gb Ethernet 1751' if=em3 drv=bnxt_en unused=igb_uio
0000:18:00.3 'BCM57504 NetXtreme-E 10Gb/25Gb/40Gb/50Gb/100Gb/200Gb Ethernet 1751' if=em4 drv=bnxt_en unused=igb_uio
0000:8a:00.0 'Ethernet Controller X710 for 10GBASE-T 15ff' if=p4p1 drv=i40e unused=igb_uio *Active*
0000:8a:00.1 'Ethernet Controller X710 for 10GBASE-T 15ff' if=p4p2 drv=i40e unused=igb_uio

Baseband devices using DPDK-compatible driver
=============================================
0000:51:00.0 'Device 0d5c' drv=igb_uio unused=

Other Baseband devices
======================
0000:52:00.0 'Device 0d5d' unused=igb_uio
0000:52:00.1 'Device 0d5d' unused=igb_uio
```

In the above example there are 2 VFs created.

- Bind with VF:

```
/var/dpdk/dpdk-21.11/usertools/dpdk-devbind.py --bind=igb_uio 52:00.0 52:00.1
```

Configure the devices using the pf_bb_config application for VF usage with both 5G and 4G (LTE) enabled.

- Select the proper config file for your test for VF:

```
./pf_bb_config ACC100 -c acc100/acc100_config_2vf_4g5g.cfg
```

- Check available interfaces and verify number of acc:

```
XR12-B:/var/pf-bb-config # lspci | grep acc
51:00.0 Processing accelerators: Intel Corporation Device 0d5c
52:00.0 Processing accelerators: Intel Corporation Device 0d5d
52:00.1 Processing accelerators: Intel Corporation Device 0d5d
```

- Test that the VF is functional on the device using bbdev-test:

```
/var/dpdk/dpdk-21.11/app/test-bbdev # /var/dpdk/dpdk-21.11/build/app/dpdk-test-bbdev -c
 F0 -a 52:00.0 -- -c validation -v ./ldpc_dec_default.data
```

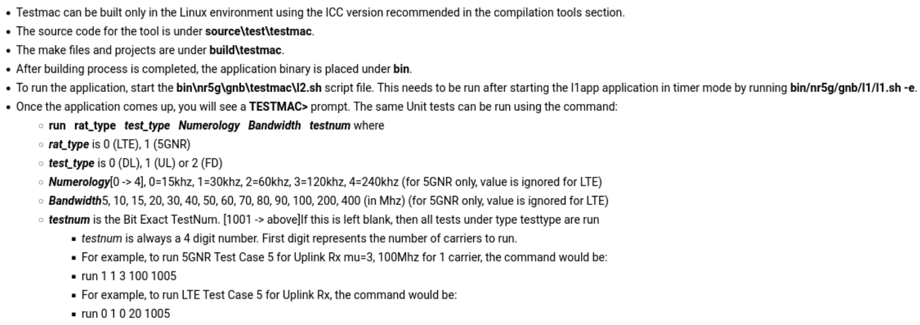
```
XR12-B:/var/dpdk/dpdk-21.11/app/test-bbdev # /var/dpdk/dpdk-21.11/build/app/dpdk-test-bbdev -c F0 -a 52:00.0 -- -c validation -v ./ldpc_dec_d
efault.data
FlexRAN SDK bblib_lte_ldpc_decoder version jenkins-FlexRAN-github-SDK-REL-113-g11d71e41
FlexRAN SDK bblib_lte_ldpc_encoder version jenkins-FlexRAN-github-SDK-REL-113-g11d71e41
FlexRAN SDK bblib_lte_LDPC_ratematch version jenkins-FlexRAN-github-SDK-REL-113-g11d71e41
FlexRAN SDK bblib_lte_rate_dematching_5gnr version jenkins-FlexRAN-github-SDK-REL-113-g11d71e41
FlexRAN SDK bblib_lte_turbo version jenkins-FlexRAN-github-SDK-REL-113-g11d71e41
FlexRAN SDK bblib_lte_crc version jenkins-FlexRAN-github-SDK-REL-113-g11d71e41
FlexRAN SDK bblib_lte_rate_matching version jenkins-FlexRAN-github-SDK-REL-113-g11d71e41
FlexRAN SDK bblib_common version jenkins-FlexRAN-github-SDK-REL-113-g11d71e41
FlexRAN SDK bblib_srs_fft_cestimate_5gnr version jenkins-FlexRAN-github-SDK-REL-113-g11d71e41
EAL: Detected CPU lcores: 40
EAL: Detected NUMA nodes: 1
EAL: Detected static linkage of DPDK
EAL: Multi-process socket /var/run/dpdk/rte/mp_socket
EAL: Selected IOVA mode 'PA'
EAL: Probe PCI driver: intel_acc100_vf (8086:d5d) device: 0000:52:00.0 (socket 0)
TELEMETRY: No legacy callbacks, legacy socket not created
WARNING: Num of operations was not provided or was set 0. Set to default (64)
WARNING: Burst size was not provided or was set 0. Set to default (32)
WARNING: Num of lcores was not provided or was set 0. Set to value from RTE config (4)

===========================================================
Starting Test Suite : BBdev Validation Tests
Test vector file = ./ldpc_dec_default.data
+ ------------------------------------------------------- +
== test: validation
dev:52:00.0, burst size: 32, num ops: 64, op type: RTE_BBDEV_OP_LDPC_DEC
Operation latency:
        avg: 40375 cycles, 25.2344 us
        min: 38332 cycles, 23.9575 us
        max: 42418 cycles, 26.5113 us
TestCase [ 0] : validation_tc passed
 + ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ +
 + Test Suite Summary : BBdev Validation Tests
 + Tests Total :        1
 + Tests Skipped :      0
 + Tests Passed :       1
 + Tests Failed :       0
 + Tests Lasted :       105.743 ms
 + ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ +
```

4 Baremetal Host Testing

4.1 FlexRAN L1 and Testmac test

Follow the steps from the **TestMac** section of *FlexRAN 5GNR Reference Solution 22.07 PHY Software Documentation* - Document #603577



- Testmac can be built only in the Linux environment using the ICC version recommended in the compilation tools section.
- The source code for the tool is under **source\test\testmac**.
- The make files and projects are under **build\testmac**.
- After building process is completed, the application binary is placed under **bin**.
- To run the application, start the **bin\nr5g\gnb\testmac\l2.sh** script file. This needs to be run after starting the l1app application in timer mode by running **bin/nr5g/gnb/l1/l1.sh -e**.
- Once the application comes up, you will see a **TESTMAC>** prompt. The same Unit tests can be run using the command:
  - **run rat_type *test_type* *Numerology* *Bandwidth* *testnum*** where
  - ***rat_type*** is 0 (LTE), 1 (5GNR)
  - ***test_type*** is 0 (DL), 1 (UL) or 2 (FD)
  - ***Numerology***[0 -> 4], 0=15khz, 1=30khz, 2=60khz, 3=120khz, 4=240khz (for 5GNR only, value is ignored for LTE)
  - ***Bandwidth***5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 400 (in Mhz) (for 5GNR only, value is ignored for LTE)
  - ***testnum*** is the Bit Exact TestNum. [1001 -> above]If this is left blank, then all tests under type testtype are run
    - *testnum* is always a 4 digit number. First digit represents the number of carriers to run.
    - For example, to run 5GNR Test Case 5 for Uplink Rx mu=3, 100Mhz for 1 carrier, the command would be:
    - run 1 1 3 100 1005
    - For example, to run LTE Test Case 5 for Uplink Rx, the command would be:
    - run 0 1 0 20 1005

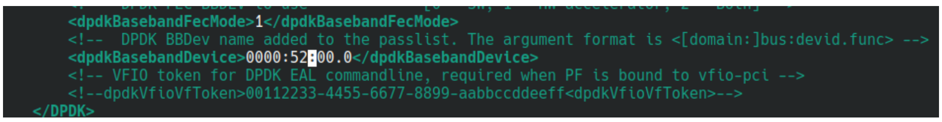
Note

Always source FlexRAN™ environment and the oneAPI in each tab and make sure that all paths are exported. For simplicity create a script to source all paths every time when running tests in each terminal.

- Change *dpdkBasebandDevice* values from phycfg_timer.xml to either physical or virtual acc:

```
/opt/FlexRan/bin/nr5g/gnb/l1 # vi phycfg_timer.xml
```

For example:



```
<dpdkBasebandFecMode>1</dpdkBasebandFecMode>
<!--  DPDK BBDev name added to the passlist. The argument format is <[domain:]bus:devid.func> -->
<dpdkBasebandDevice>0000:52:00.0</dpdkBasebandDevice>
<!-- VFIO token for DPDK EAL commandline, required when PF is bound to vfio-pci -->
<!--dpdkVfioVfToken>00112233-4455-6677-8899-aabbccddeeff<dpdkVfioVfToken>-->
</DPDK>
```

Where FecMode is set to 1 (HW accelertor) and 0000:52:00.0 is the VF value from acc. Set *dpdkBasebandFecMode* to VF value according to your specific card.

- From terminal 1 run:

```
./FlexRAN-<version>/bin/nr5g/gnb/l1/l1.sh -e
```

You should be able to see the following console:

```
=====================================================================================================
Non BBU threads in application
=====================================================================================================
phy_print_thread:                      [PID:  29297] binding on [CPU  0] [PRIO:  0] [POLICY:  1]
wls_rx_handler (non-rt):                [PID:  29301] binding on [CPU  0]
=====================================================================================================

PHY>welcome to application console
```

- From the 2nd terminal run:

```
/var/FlexRan22.07/bin/nr5g/gnb/testmac # ./l2.sh

run 1 1 3 100 1005
```

```
        4065  4066  4071  4072  4073  4074
TESTMAC>welcome to application console


TESTMAC>run 1 1 3 100 1005
```

See examples from *FlexRAN 5GNR Reference Solution 22.07 PHY Software Documentation* Document #603577 **TestMac** section:

- To run the application, start the **bin\nr5g\gnb\testmac\l2.sh** script file. This needs to be run after starting the l1app application in timer mode by running **bin/nr5g/gnb/l1/l1.sh -e**.
- Once the application comes up, you will see a **TESTMAC>** prompt. The same Unit tests can be run using the command:
  - **run rat_type *test_type* *Numerology* *Bandwidth* *testnum*** where
  - ***rat_type*** is 0 (LTE), 1 (5GNR)
  - ***test_type*** is 0 (DL), 1 (UL) or 2 (FD)
  - ***Numerology***[0 -> 4], 0=15khz, 1=30khz, 2=60khz, 3=120khz, 4=240khz (for 5GNR only, value is ignored for LTE)
  - ***Bandwidth***5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 400 (in Mhz) (for 5GNR only, value is ignored for LTE)
  - ***testnum*** is the Bit Exact TestNum. [1001 -> above]If this is left blank, then all tests under type testtype are run
    - *testnum* is always a 4 digit number. First digit represents the number of carriers to run.
    - For example, to run 5GNR Test Case 5 for Uplink Rx mu=3, 100Mhz for 1 carrier, the command would be:
    - run 1 1 3 100 1005
    - For example, to run LTE Test Case 5 for Uplink Rx, the command would be:
    - run 0 1 0 20 1005

The connection should be established in the 1st terminal once you'll run l2.sh from the 2nd terminal:

```
timer_reg_proc_symbol: nSymbol: 6, pProc: 0x8d2570, lpData: (nil)
timer_reg_proc_symbol: nSymbol: 10, pProc: 0x8d25a0, lpData: (nil)
timer_reg_proc_symbol: nSymbol: 1, pProc: 0x8d25d0, lpData: (nil)
timer_reg_proc_symbol: nSymbol: 3, pProc: 0x8d2600, lpData: (nil)
nr5g_gnb_urllc_register_call_backs: nTimerMode[1] nUrllcMiniSlotMask[0]
timer_main_thread:        [PID:  29386] binding on [CPU  2] [PRIO: 96] [POLICY:  1]
                          Numerology: [     3], numSlotsPerSubframe: [ 8], ttiPeriod: [ 125 usecs]
                         Period (usecs):    357    714   1071   1428   1785   2142   2500   2857   3214   3571   3928
4285   4642   5000
                     Fn Callbacks (Sym):      0      1      2      3      4      5      6      7      8      9     10
  11     12     13
                          Instance 0  :    YES    YES     NO    YES     NO     NO    YES     NO     NO     NO    YES
  NO     NO     NO
                          Instance 1  :     NO     NO     NO     NO     NO     NO     NO     NO     NO     NO     NO
  NO     NO     NO
                          Instance 2  :     NO     NO     NO     NO     NO     NO     NO     NO     NO     NO     NO
  NO     NO     NO

ebbu_pool_update_frame_slot_sym_num: PhyId[0] nSlotIdx[7997] frame,slot[1023,78] gNumSlotPerSfn[80]
ebbu_pool_update_multi_cell_status Call Stop: (PhyStartCurrCount 17 PhyStartCount 17)
phydi_stop[from 2]: phyInstance: -1, sendStop: 1, phyIdStart: 0, phyIdStop: 1
    PHY_STOP PhyInstance[0] PhyState[1] PhyStartMode[1] PhyStateCount[1]
```

In the 2nd terminal you should be able to see test result:

```
wls_mac_print_stats:
        nTotalBlocks[4009] nAllocBlocks[2399] nFreeBlocks[1610] nWaterMarkAllocBlocks[2418]
        nTotalAllocCnt[2509] nTotalFreeCnt[110] Diff[2399]
        nDlBufAllocCnt[57] nDlBufFreeCnt[57] Diff[0]
        nUlBufAllocCnt[2452] nUlBufFreeCnt[53] Diff[2399]

All Tests Completed, Total run 1 Tests, PASS 1 Tests, and FAIL 0 Tests
----------------------------------------------------------------------------
mem_mgr_display_size:
    Num Memory Alloc:             11
    Total Memory Size:        264,018
----------------------------------------------------------------------------
```

Another test case is to use a preconfigured test file.

For example from the 2nd terminal run:

```
/var/FlexRan22.07/bin/nr5g/gnb/testmac # ./l2.sh --testfile=/var/FlexRan22.07/bin/nr5g/
gnb/testmac/icelake-sp/icxsp_mu1_100mhx_4x4_hton.cfg
```

```
-----------------------------------
              |  MAC |    MAC-to-PHY Tput    |              PHY-to-MAC Tput              |  UL FEC CB Iteration
s |
      Cell    | Inst |    kbps       Num CB |          kbps         UL BLER       Num CB |    Min     Avg    Max
 |      SRS SNR
 -------------|------|---------------------|-----------------------------------------|---------------------
--|-------------
      0 (MU 1) |   0  |   863,091     567,928 |    72,960 /    72,960      0.00%      63,968 |      1   1.00      1
 |      0 Db
----------------------------------------------------------------------------------------------------------------
-----------------------------------
Core Utilization [2 BBU core(s)]:
    Core Id :      4    36     Avg
  Numa Node :      0     0
     Util % :  34.04 34.06   34.05
     Intr % :   0.68  0.69    0.69
    Spare % :   0.62  0.62    0.62
    Sleep % :  64.64 64.62   64.63
    TTI Cnt :   8200  8200
    TTI Min :      1     0
    TTI Avg :     33    33
    TTI Max :     96    96
----------------------------------------------------------------------------------------------------------------
```

```
~~~~--------~~~~--------~~~~------~~~~~~-
==== l1app [Time:    0Hr  0Min 20Sec ] NumCarrier: 2 NumBbuCores: 3. Tti2Tti Time: [500.00..504.15..510.00] usces
----------------------------------------------------------------------------------------------------------------
--------------------------------
                  |                usecs              |            % of TTI
    Latency       |      Min        Avg         Max |       Min         Avg         Max
    --------------|---------------------------------|-----------------------------
    DL_LINK  MU1  |     125.00     202.21     385.00 |        25%         40%         77%
    UL_LINK  MU1  |     860.00     875.67     890.00 |       172%        175%        178%
    SRS_LINK MU1  |       0.00       0.00       0.00 |         0%          0%          0%
----------------------------------------------------------------------------------------------------------------
--------------------------------
              |  MAC |    MAC-to-PHY Tput    |              PHY-to-MAC Tput              |  UL FEC CB Iterations
 |
      Cell    | Inst |    kbps       Num CB |          kbps         UL BLER       Num CB |    Min     Avg    Max
 |      SRS SNR
 -------------|------|---------------------|-----------------------------------------|---------------------
-|-------------
      0 (MU 1) |   0  |   300,851     255,968 |    28,672 /    28,672      0.00%      31,984 |      1   1.00      1
 |      0 Db
      1 (MU 1) |   0  |   300,851     255,968 |    28,672 /    28,672      0.00%      31,984 |      1   1.00      1
 |      0 Db
----------------------------------------------------------------------------------------------------------------
--------------------------------
```

```
----------------------------------------------------------------------------
mem_mgr_display_size:
    Num Memory Alloc:             12
    Total Memory Size:        268,818
----------------------------------------------------------------------------


Test[FD_mu1_100mhz_4308] Completed
wls_mac_print_stats:
        nTotalBlocks[4009] nAllocBlocks[2399] nFreeBlocks[1610] nWaterMarkAllocBlocks[2874]
        nTotalAllocCnt[82221232] nTotalFreeCnt[82218833] Diff[2399]
        nDlBufAllocCnt[64873831] nDlBufFreeCnt[64873831] Diff[0]
        nUlBufAllocCnt[17347401] nUlBufFreeCnt[17345002] Diff[2399]

All Tests Completed, Total run 27 Tests, PASS 21 Tests, and FAIL 6 Tests
----------------------------------------------------------------------------
mem_mgr_display_size:
    Num Memory Alloc:             11
    Total Memory Size:        264,018
----------------------------------------------------------------------------
```

Note

Number of failed tests listed above related to a different number of CPU cores defined in the test file.

4.2 CPU set shielding

Another tool for more tuned cores isolation is cpu set shielding.

You can also review CPU manipulation commands from → https://documentation.suse.com/sle-rt/15-SP3/pdf/book-shielding_color_en.pdf

Some examples of using shielding on CPU with integrated tools like cset.

- Create a cset called flexran_set

```
XR12-B:/var/FlexRan22.07/bin/nr5g/gnb/testmac/icelake-sp # cset set -c 7-11 -s flexran_set
cset: --> created cpuset "flexran_set"
XR12-B:/var/FlexRan22.07/bin/nr5g/gnb/testmac/icelake-sp # cset set -l
cset:
         Name       CPUs-X    MEMs-X Tasks Subs Path
 ------------ ---------- - ------- - ----- ---- ----------
         root       0-39 y       0 y  1329    3 /
         user       6-10 n       0 n     0    0 /user
  flexran_set       7-11 n       0 n     0    0 /flexran_set
       system        0-5 n       0 n     0    0 /system
```

Example of moving *top* command from root set to flexran_set:

```
XR12-B:/var/FlexRan22.07/bin/nr5g/gnb/testmac/icelake-sp # cset proc -l -s root | grep 6135
 root       6135  6103 Soth top
 root       6885 35056 Soth grep --color=auto 6135
XR12-B:/var/FlexRan22.07/bin/nr5g/gnb/testmac/icelake-sp # cset proc -m -p 6135 -t flexran_set
cset: moving following pidspec: 6135
cset: moving 1 userspace tasks to /flexran_set
cset: done
XR12-B:/var/FlexRan22.07/bin/nr5g/gnb/testmac/icelake-sp # cset set -l -s flexran_set
cset:
         Name       CPUs-X    MEMs-X Tasks Subs Path
 ------------ ---------- - ------- - ----- ---- ----------
  flexran_set       7-11 n       0 n     1    0 /flexran_set
XR12-B:/var/FlexRan22.07/bin/nr5g/gnb/testmac/icelake-sp # cset proc -l -s flexran_set
cset: "flexran_set" cpuset of CPUSPEC(7-11) with 1 task running
 USER       PID  PPID SPPr TASK NAME
 -------- ----- ----- ---- ---------
 root      6135  6103 Soth top
```

When starting a testmac you can move pid to a dedicated cset:

```
XR12-B:~ # cset proc -m -p 9319,9444 -t flexran_set
cset: moving following pidspec: 9319,9444
cset: moving 2 userspace tasks to /flexran_set
cset: done
```

```
XR12-B:~ # cset proc -l -s flexran_set
cset: "flexran_set" cpuset of CPUSPEC(7-11) with 2 tasks running
 USER       PID  PPID SPPr TASK NAME
 -------- ----- ----- ---- ---------
 root      9319  9304 Soth ./l1app --cfgfile=phycfg_timer.xml
 root      9444  9436 Soth ./testmac DIR_WIRELESS_TEST_4G=/var/FlexRan22.07/...
```

To move all siblings from pid use -–*threads* option:

```
cset proc -m -p 16165 --threads -t two
```

For all features of CPU manipulations please review shielding tasks documents for CPU isolations:

- https://www.suse.com/c/cpu-isolation-introduction-part-1/
- https://documentation.suse.com/sle-rt/15-SP3/pdf/book-shielding_color_en.pdf
- https://documentation.suse.com/sle-rt/15-SP3/pdf/article-virtualization_color_en.pdf#%5B%7B%22num%22%3A30%2C%22gen%22%3A0%7D%2C%7B%22name%22%3A%22XYZ%22%7D%2C63.779%2C788.031%2Cnull%5D

To run Testmac with VF set, change setting to proper VF value and configuration

```
XR12-B:/var/pf-bb-config # ./pf_bb_config ACC100 -c acc100/acc100_config_vf_5g.cfg
== pf_bb_config Version #VERSION_STRING# ==
Queue Groups: 4 5GUL, 4 5GDL, 0 4GUL, 0 4GDL
Number of 5GUL engines 8
Configuration in VF mode
 ROM version MM 99AD92
PF ACC100 configuration complete
ACC100 PF [0000:51:00.0] configuration complete!
```

and from the 1st terminal run:

```
./l1.sh -e
```

From the 2nd terminal run:

```
run 1 1 3 100 1005
```

```
---------------------------------------------------------------------------


1005 | Result: PASS | DL_IQ: - | PUSCH: P | RXBITS: - | PUCCH: - | MUXSCH: - | SNR: P | TA: P | RACH: - | SRS: - | RI
P: - | NFLOOR: - | NDEMOD_CW: - | DL_BW: - | UL_BW: - |


wls_mac_print_stats:
        nTotalBlocks[4009] nAllocBlocks[2399] nFreeBlocks[1610] nWaterMarkAllocBlocks[2417]
        nTotalAllocCnt[2509] nTotalFreeCnt[110] Diff[2399]
        nDlBufAllocCnt[57] nDlBufFreeCnt[57] Diff[0]
        nUlBufAllocCnt[2452] nUlBufFreeCnt[53] Diff[2399]

All Tests Completed, Total run 1 Tests, PASS 1 Tests, and FAIL 0 Tests
---------------------------------------------------------------------------
mem_mgr_display_size:
    Num Memory Alloc:              11
    Total Memory Size:         264,018
---------------------------------------------------------------------------
```

```
XR12-B:~ # cset proc -m -p 13726,13849 --threads -t flexran_set
cset: moving following pidspec: 13726,13849,13727,13728,13729,13765,13766,13887,13888,13889,13890,13891
cset: moving 12 userspace tasks to /flexran_set
[==================================================]%
cset: done
```

```
XR12-B:~ # cset proc -l -s flexran_set
cset: "flexran_set" cpuset of CPUSPEC(7-11) with 12 tasks running
 USER       PID  PPID SPPr TASK NAME
 -------- ----- ----- ---- ---------
 root     13726 13711 Soth ./l1app --cfgfile=phycfg_timer.xml
 root     13727 13711 Soth ./l1app --cfgfile=phycfg_timer.xml
 root     13728 13711 Soth ./l1app --cfgfile=phycfg_timer.xml
 root     13729 13711 Soth ./l1app --cfgfile=phycfg_timer.xml
 root     13765 13711 Soth ./l1app --cfgfile=phycfg_timer.xml
 root     13766 13711 Soth ./l1app --cfgfile=phycfg_timer.xml
 root     13849 13841 Soth ./testmac DIR_WIRELESS_TEST_4G=/var/FlexRan22.07/...
 root     13887 13841 Soth ./testmac DIR_WIRELESS_TEST_4G=/var/FlexRan22.07/...
 root     13888 13841 Soth ./testmac DIR_WIRELESS_TEST_4G=/var/FlexRan22.07/...
 root     13889 13841 Soth ./testmac DIR_WIRELESS_TEST_4G=/var/FlexRan22.07/...
 root     13890 13841 Sf90 ./testmac DIR_WIRELESS_TEST_4G=/var/FlexRan22.07/...
 root     13891 13841 Sf89 ./testmac DIR_WIRELESS_TEST_4G=/var/FlexRan22.07/...
```

If using a config file, from the 2nd terminal run:

```
./l2.sh --testfile=/var/FlexRan22.07/bin/nr5g/gnb/testmac/icelake-sp/
icxsp_mu1_100mhz_mmimo_32x32_hton.cfg
```

If using a *taskset*, from terminal 1 run:

```
~/gnb/l1 # taskset -c 12-19 ./l1.sh -e
```

and from terminal 2 run:

```
~/gnb/testmac # taskset -c 12-19 ./l2.sh
```

5 Deploy FlexRAN™ on Container through Kubernetes

5.1 Generate LTE/5G Docker Images with pre-build FlexRAN™

All prerequisite components and FlexRAN™ should be installed as descrtibe in the previous sections.

The main document to follow: FlexRAN Reference Solution Cloud-Native Setup (Intel® Document Number: 575834-15.0)

Use existing FlexRAN directory or create a FlexRAN pre-configured directory which will be used for the container image.

Source all environment variables:

```
export RTE_SDK=/var/dpdk/dpdk-21.11
source /opt/intel/oneapi/setvars.sh
export PKG_CONFIG_PATH=$DIR_WIRELESS_SDK/pkgcfg:$PKG_CONFIG_PATH
source set_env_var.sh
```

5.1.1 Create a Dockerfile

If you want to deploy a SUSE Linux Enterprise-based container to deploy to the cluster in the future, follow the steps below.

- Modify *flexran_build_dockerfile.sh* from the flexran directory:

```
OS_TYPE_sle=`ls /boot/efi/EFI/ | grep sle`
if [ -n "$OS_TYPE_sle" ]; then
  if [ -z $http_proxy ];then
      cat > flexran_build/Dockerfile << EOF
FROM registry.suse.com/suse/sle15:15.3
ENV no_proxy "localhost,127.0.0.1,192.168.0.100"
ADD http://192.168.150.160/repo/rmt-server.crt /etc/pki/trust/anchors/rmt.crt
ARG ADDITIONAL_MODULES
RUN update-ca-certificates
 RUN zypper --gpg-auto-import-keys ref -s
 RUN zypper ref && zypper --non-interactive in libhugetlbfs libhugetlbfs-devel gcc11-c++ numactl ethto
ol gcc make kmod wget patch iproute2 pciutils python vim cmake unzip iputils libaio1 libaio-devel git
git-core net-tools gawk
ENV WIRELESS_SDK_TARGET_ISA=avx512 CPA_DIR=/opt/flexran/libs/cpa XRAN_DIR=/opt/flexran/xran DIR_WIRELE
SS_SDK_ROOT=/opt/flexran/sdk SDK_BUILD=build-avx512-icx DIR_WIRELESS_SDK=/opt/flexran/sdk/build-avx512
-icx FLEXRAN_SDK=/opt/flexran/sdk/build-avx512-icx/install DIR_WIRELESS_FW=/opt/flexran/framework DIR_
WIRELESS_TEST_4G=/opt/flexran/tests/lte DIR_WIRELESS_TEST_5G=/opt/flexran/tests/nr5g DIR_WIRELESS_TABL
E_5G=/opt/flexran/bin/nr5g/gnb/l1/table
WORKDIR /var/
COPY flexran ./flexran
COPY oneapi /opt/oneapi
COPY docker_entry.sh ./
EOF
  else
      cat > $tmp_path/Dockerfile << EOF
FROM registry.suse.com/suse/sle15:15.3
# ENV http_proxy $http_proxy
# ENV https_proxy $https_proxy
RUN zypper ref && zypper --non-interactive in libhugetlbfs libhugetlbfs-devel numactl ethtool gcc make
 kmod wget patch iproute2 pciutils python vim cmake unzip iputils libaio1 libaio-devel git git-core ne
t-tools gawk
ENV WIRELESS_SDK_TARGET_ISA=avx512 CPA_DIR=/opt/flexran/libs/cpa XRAN_DIR=/opt/flexran/xran DIR_WIRELE
SS_SDK_ROOT=/opt/flexran/sdk SDK_BUILD=build-avx512-icx DIR_WIRELESS_SDK=/opt/flexran/sdk/build-avx512
-icx FLEXRAN_SDK=/opt/flexran/sdk/build-avx512-icx/install DIR_WIRELESS_FW=/opt/flexran/framework DIR_
WIRELESS_TEST_4G=/opt/flexran/tests/lte DIR_WIRELESS_TEST_5G=/opt/flexran/tests/nr5g DIR_WIRELESS_TABL
E_5G=/opt/flexran/bin/nr5g/gnb/l1/table

WORKDIR /var/
```

Note

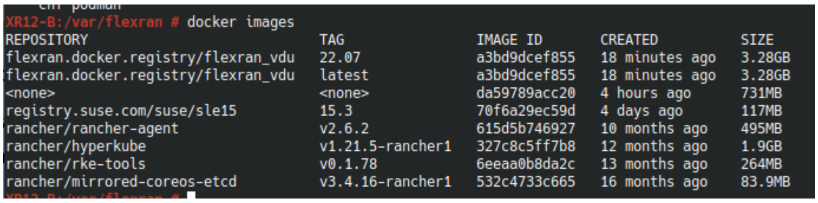
Modify according to your local setup. If local RMT server is used, you need to post rmt-server.crt file on your RMT server in the location which can be reachable from url. So, on the local RMT server copy */etc/rmt/ssl/rmt-server.crt* file to the */usr/share/rmt/public/repo* directory, which creates symb link to *./var/lib/rmt/public/repo* which is a public repo of RMT server. Setup a proper permission to */usr/share/rmt/public/repo* directory. Sync rmt server.

- Build a docker image:

```
./flexran_build_dockerfile_suse.sh -v -e avx512 -r 5gnr -m all –x icx
```

- Tag a docker image:

```
docker tag flexran.docker.registry/flexran_vdu:latest flexran.docker.registry/
flexran_vdu:22.07
```



Note

Another alternative and recommended tool to use is podman since it's daemonless and has integration with cockpit web console on Sle Micro. For that you need to replace *docker build* command with *podman* in flexran_build_dockerfile.sh file and run:

```
podman build -t
```

For more details review the Podman guide: https://documentation.suse.com/sle-micro/5.1/pdf/article-podman_color_en.pdf

- Prepare file to export to the target node and save docker as:

```
docker save flexran.docker.registry/flexran_vdu:22.07|gzip > flexranimage.tar.gz
```

5.2 Create an RKE2 cluster

5.2.1 Install SUSE Linux Enterprise Micro

In this test deployment, SUSE Linux Enterprise Micro 5.2 (SLE Micro) was used as a server host for the Rancher server test deployment.

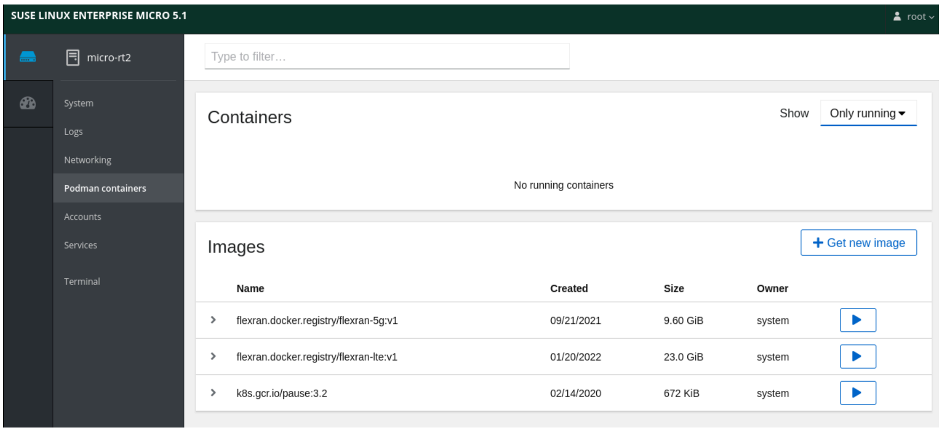
SUSE Linux Enterprise Micro is a lightweight and secure OS platform purpose built for containerized and virtualized workloads.

For more details on installation of SLE Micro, review: https://documentation.suse.com/sle-micro/5.2/pdf/book-deployment-slemicro_color_en.pdf

After installing a SLE Micro you can enable a cockpit console for easy management:

```
systemctl enable --now cockpit.socket
```

and open console in the browser as 'https://your-ip:9090/'



For more details review:

https://documentation.suse.com/sle-micro/5.2/

5.2.2 Install a Rancher server

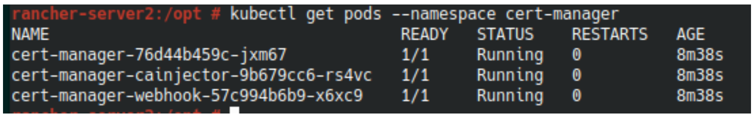
`Install K3s:`

```
curl -sfL https://get.k3s.io | INSTALL_K3S_VERSION="v1.23.9+k3s1"
 INSTALL_K3S_SKIP_SELINUX_RPM=true INSTALL_K3S_EXEC='server --cluster-init --write-
kubeconfig-mode=644' sh -s -
```

Install certificates and verify:

```
kubectl apply --validate=false -f https://github.com/cert-manager/cert-manager/releases/
download/v1.7.1/cert-manager.crds.yaml
 helm repo add jetstack https://charts.jetstack.io
 helm repo update
 export KUBECONFIG=/etc/rancher/k3s/k3s.yaml
helm install cert-manager jetstack/cert-manager --namespace cert-manager --create-
namespace --version v1.7.1
```

```
kubectl get pods --namespace cert-manager
```



Install Rancher:

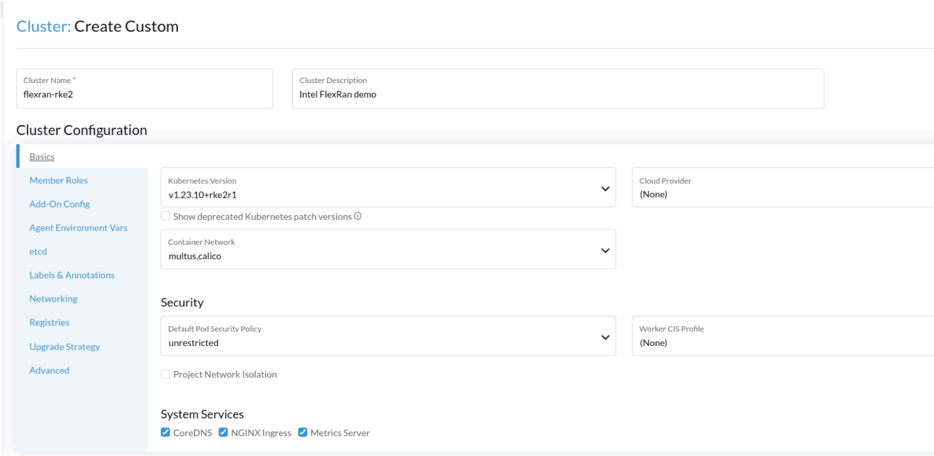
```
helm repo add rancher-stable https://releases.rancher.com/server-charts/stable
kubectl create namespace cattle-system
export HOSTNAME="rancher-server2.isv.suse"
export RANCHER_VERSION="2.6.5"
helm install rancher rancher-stable/rancher --namespace cattle-system --set
 hostname=rancher-server2.isv.suse --set version=2.6.5 --set replicas=1
```

Go to Rancher url and login.

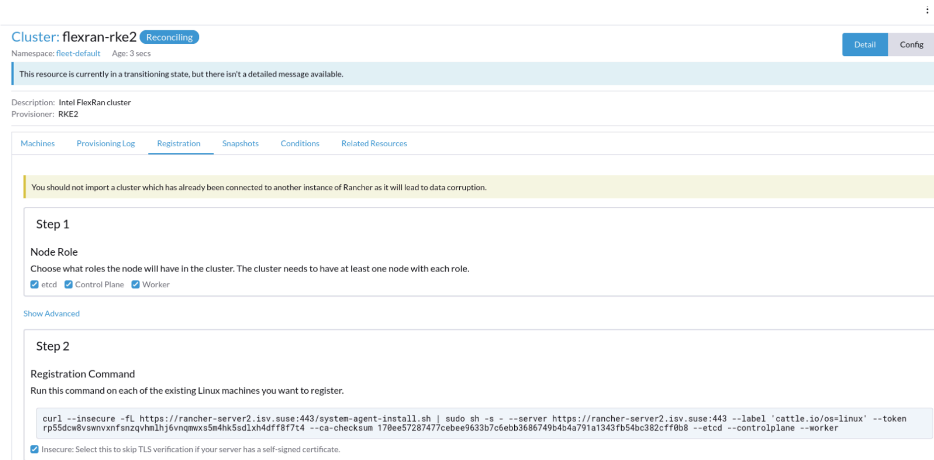
For more details on Rancher installation, review > https://documentation.suse.com/trd/kubernetes/pdf/kubernetes_ri_rancher-k3s-slemicro_color_en.pdf

5.2.3 Create a custom RKE2 cluster

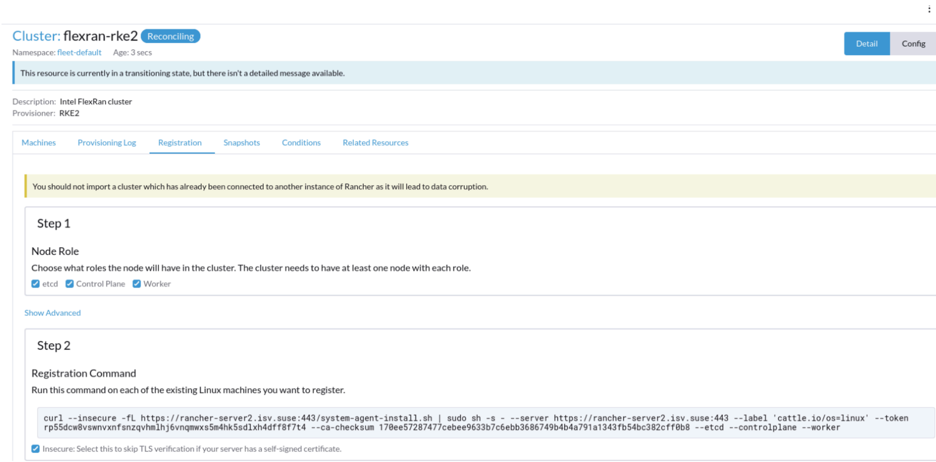
- From the Rancher server create a custom cluster > switch to rke2



- Copy registration script to a new node to add it to the cluster:



- Verify if machines got provisioned:




```
kubectl get nodes
NAME     STATUS   ROLES                              AGE     VERSION
xr12-a   Ready    control-plane,etcd,master,worker   7d21h   v1.23.10+rke2r1
xr12-c   Ready    control-plane,etcd,master,worker   7d20h   v1.23.10+rke2r1
```

In this test case 2 Dell XR12 nodes were used with Sle Micro 5.2 RT installed as part of the RKE2 cluster. Both target nodes should have dpdk with a patch and Intel oneAPI installed.

For core isolation on Sle Micro RT, install *tuned* package with additional dependencies.

```
transactional-update pkg install tuned.rpm python3-configobj.rpm python3-linux-procfs.rpm
 python3-pyudev.rpm virt-what.rpm
```

Note

For this test, SLES 15 repositories were used with *curl* commands to download packages locally. For a large scale deployment a local repository can be made with required RPMs.

Modify */etc/default/grub* to the required tuned parameters with isolcpu and run transactional-update grub.cfg to save changes and reboot.

Note

When setting up CPU Manager for Kubernetes* (CMK*) it should be based on isolcpu settings in GRUB. Make sure that all required plugins for Kubernetes for your test are installed on tested nodes as described in section 4 of Intel's document 575834-15.0

> **Note**
>
> It's not recommended to add a FlexRAN™ development node to the RKE2 cluster. Instead, move image to the FlexRAN™ RKE2 cluster, either manually or with a repo.

During our RKE2 cluster deployment, Rancher provides an option to select Multus and Calico as default plugins, so no needs to install them manually.

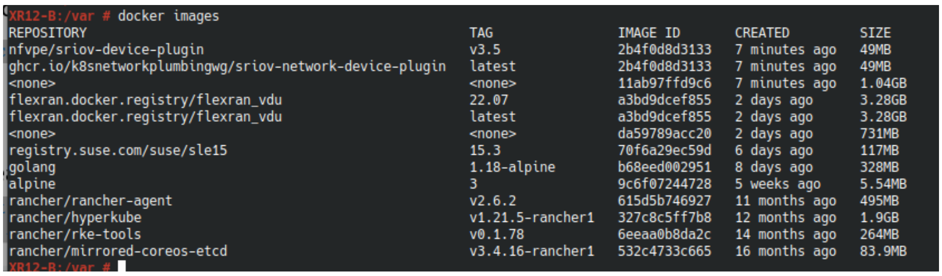
5.3 Build SR-IOV Network Device Plugin

The setup details for virtual or physical functions of the SR-IOV Network Device Plugins can be found at: https://github.com/k8snetworkplumbingwg/sriov-network-device-plugin

```
cd /root/go/src/github.com/intel/
~/go/src/github.com/intel # git clone https://github.com/intel/sriov-network-device-plugin
cd sriov-network-device-plugin/
git checkout v3.5.1
mkdir bin
cp ~/go/bin/golint bin/
~/go/src/github.com/intel/sriov-network-device-plugin # make
make image
```

Tag with:

```
docker tag ghcr.io/k8snetworkplumbingwg/sriov-network-device-plugin:latest nfvpe/sriov-device-plugin:v3.5
```



```
XR12-B:/var # docker images
REPOSITORY                                               TAG                  IMAGE ID       CREATED          SIZE
nfvpe/sriov-device-plugin                                v3.5                 2b4f0d8d3133   7 minutes ago    49MB
ghcr.io/k8snetworkplumbingwg/sriov-network-device-plugin latest               2b4f0d8d3133   7 minutes ago    49MB
<none>                                                   <none>               11ab97ffd9c6   7 minutes ago    1.04GB
flexran.docker.registry/flexran_vdu                      22.07                a3bd9dcef855   2 days ago       3.28GB
flexran.docker.registry/flexran_vdu                      latest               a3bd9dcef855   2 days ago       3.28GB
<none>                                                   <none>               da59789acc20   2 days ago       731MB
registry.suse.com/suse/sle15                             15.3                 70f6a29ec59d   6 days ago       117MB
golang                                                   1.18-alpine          b68eed002951   8 days ago       328MB
alpine                                                   3                    9c6f07244728   5 weeks ago      5.54MB
rancher/rancher-agent                                    v2.6.2               615d5b746927   11 months ago    495MB
rancher/hyperkube                                        v1.21.5-rancher1     327c8c5ff7b8   12 months ago    1.9GB
rancher/rke-tools                                        v0.1.78              6eeaa0b8da2c   14 months ago    264MB
rancher/mirrored-coreos-etcd                             v3.4.16-rancher1     532c4733c665   16 months ago    83.9MB
XR12-B:/var #
```

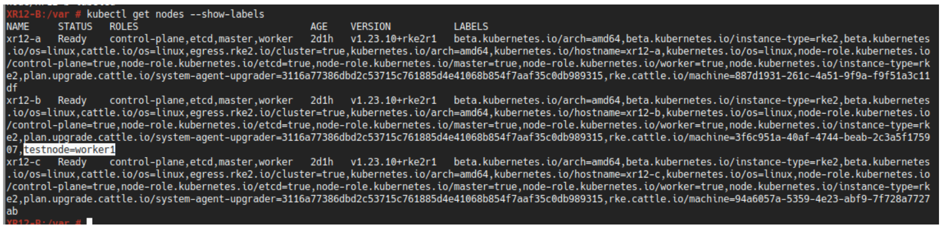
Save with:

```
docker save nfvpe/sriov-device-plugin:v3.5|gzip > sriov-device-plugin.tar.gz
```

5.4 Create FlexRAN™ Pods

Label nodes as:

```
kubectl label nodes xr12-b testnode=worker1
```



- Configure FEC and FVL SRIOV

To reconfigure pf_bb_config run:

```
pkill pf_bb_config
modprobe vfio-pci enable_sriov=1 disable_idle_d3=1
insmod /var/dpdk/dpdk-kmods/linux/igb_uio/igb_uio.ko
/var/dpdk/dpdk-21.11/usertools/dpdk-devbind.py -b igb_uio 18:00.0
```

where 18:00.0 is acc pf address

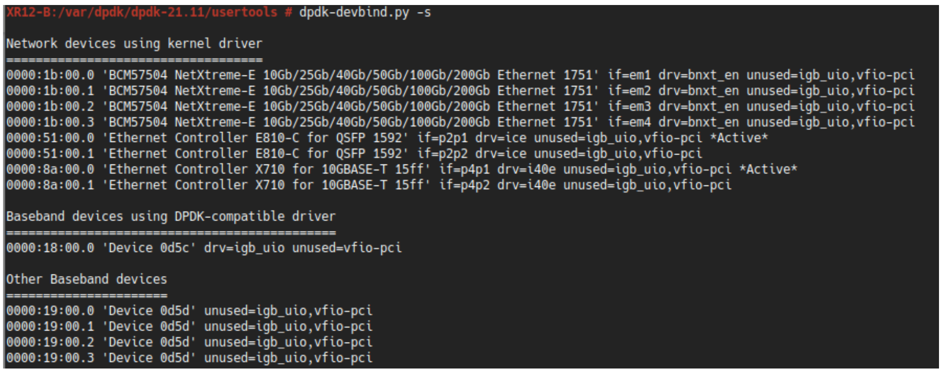
Check available accelerator cards:

```
lspci|grep acc
18:00.0 Processing accelerators: Intel Corporation Device 0d5c
```

Add 4 VFs to acc:

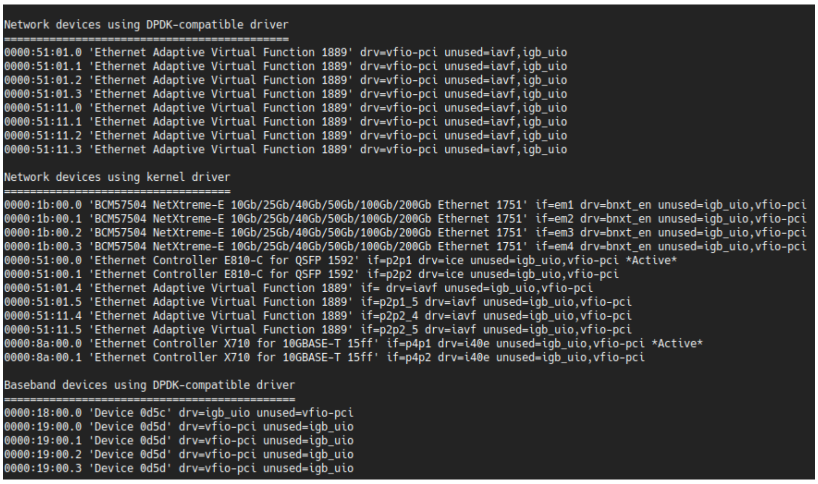
```
echo 4 > /sys/bus/pci/devices/0000:18:00.0/max_vfs
```

Verify:



In the below example 4 new were created:

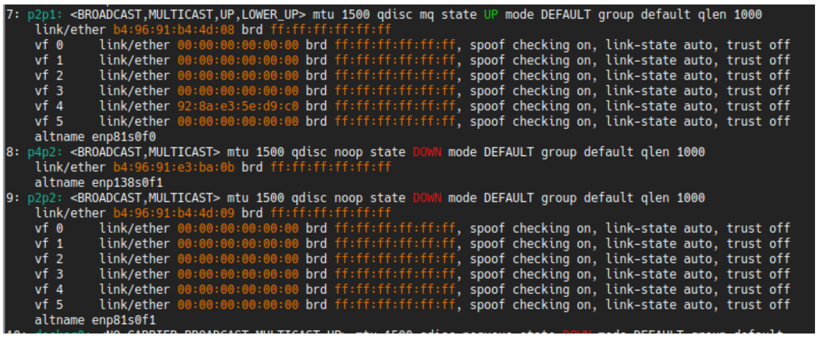
```
lspci|grep acc
18:00.0 Processing accelerators: Intel Corporation Device 0d5c
19:00.0 Processing accelerators: Intel Corporation Device 0d5d
19:00.1 Processing accelerators: Intel Corporation Device 0d5d
19:00.2 Processing accelerators: Intel Corporation Device 0d5d
19:00.3 Processing accelerators: Intel Corporation Device 0d5d
```



```
Network devices using DPDK-compatible driver
============================================
0000:51:01.0 'Ethernet Adaptive Virtual Function 1889' drv=vfio-pci unused=iavf,igb_uio
0000:51:01.1 'Ethernet Adaptive Virtual Function 1889' drv=vfio-pci unused=iavf,igb_uio
0000:51:01.2 'Ethernet Adaptive Virtual Function 1889' drv=vfio-pci unused=iavf,igb_uio
0000:51:01.3 'Ethernet Adaptive Virtual Function 1889' drv=vfio-pci unused=iavf,igb_uio
0000:51:11.0 'Ethernet Adaptive Virtual Function 1889' drv=vfio-pci unused=iavf,igb_uio
0000:51:11.1 'Ethernet Adaptive Virtual Function 1889' drv=vfio-pci unused=iavf,igb_uio
0000:51:11.2 'Ethernet Adaptive Virtual Function 1889' drv=vfio-pci unused=iavf,igb_uio
0000:51:11.3 'Ethernet Adaptive Virtual Function 1889' drv=vfio-pci unused=iavf,igb_uio

Network devices using kernel driver
===================================
0000:1b:00.0 'BCM57504 NetXtreme-E 10Gb/25Gb/40Gb/50Gb/100Gb/200Gb Ethernet 1751' if=em1 drv=bnxt_en unused=igb_uio,vfio-pci
0000:1b:00.1 'BCM57504 NetXtreme-E 10Gb/25Gb/40Gb/50Gb/100Gb/200Gb Ethernet 1751' if=em2 drv=bnxt_en unused=igb_uio,vfio-pci
0000:1b:00.2 'BCM57504 NetXtreme-E 10Gb/25Gb/40Gb/50Gb/100Gb/200Gb Ethernet 1751' if=em3 drv=bnxt_en unused=igb_uio,vfio-pci
0000:1b:00.3 'BCM57504 NetXtreme-E 10Gb/25Gb/40Gb/50Gb/100Gb/200Gb Ethernet 1751' if=em4 drv=bnxt_en unused=igb_uio,vfio-pci
0000:51:00.0 'Ethernet Controller E810-C for QSFP 1592' if=p2p1 drv=ice unused=igb_uio,vfio-pci *Active*
0000:51:00.1 'Ethernet Controller E810-C for QSFP 1592' if=p2p2 drv=ice unused=igb_uio,vfio-pci
0000:51:01.4 'Ethernet Adaptive Virtual Function 1889' if= drv=iavf unused=igb_uio,vfio-pci
0000:51:01.5 'Ethernet Adaptive Virtual Function 1889' if=p2p1_5 drv=iavf unused=igb_uio,vfio-pci
0000:51:11.4 'Ethernet Adaptive Virtual Function 1889' if=p2p2_4 drv=iavf unused=igb_uio,vfio-pci
0000:51:11.5 'Ethernet Adaptive Virtual Function 1889' if=p2p2_5 drv=iavf unused=igb_uio,vfio-pci
0000:8a:00.0 'Ethernet Controller X710 for 10GBASE-T 15ff' if=p4p1 drv=i40e unused=igb_uio,vfio-pci *Active*
0000:8a:00.1 'Ethernet Controller X710 for 10GBASE-T 15ff' if=p4p2 drv=i40e unused=igb_uio,vfio-pci

Baseband devices using DPDK-compatible driver
=============================================
0000:18:00.0 'Device 0d5c' drv=igb_uio unused=vfio-pci
0000:19:00.0 'Device 0d5d' drv=vfio-pci unused=igb_uio
0000:19:00.1 'Device 0d5d' drv=vfio-pci unused=igb_uio
0000:19:00.2 'Device 0d5d' drv=vfio-pci unused=igb_uio
0000:19:00.3 'Device 0d5d' drv=vfio-pci unused=igb_uio
```



```
7: p2p1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000
    link/ether b4:96:91:b4:4d:08 brd ff:ff:ff:ff:ff:ff
    vf 0     link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff, spoof checking on, link-state auto, trust off
    vf 1     link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff, spoof checking on, link-state auto, trust off
    vf 2     link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff, spoof checking on, link-state auto, trust off
    vf 3     link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff, spoof checking on, link-state auto, trust off
    vf 4     link/ether 92:8a:e3:5e:d9:c0 brd ff:ff:ff:ff:ff:ff, spoof checking on, link-state auto, trust off
    vf 5     link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff, spoof checking on, link-state auto, trust off
    altname enp81s0f0
8: p4p2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether b4:96:91:e3:ba:0b brd ff:ff:ff:ff:ff:ff
    altname enp138s0f1
9: p2p2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether b4:96:91:b4:4d:09 brd ff:ff:ff:ff:ff:ff
    vf 0     link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff, spoof checking on, link-state auto, trust off
    vf 1     link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff, spoof checking on, link-state auto, trust off
    vf 2     link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff, spoof checking on, link-state auto, trust off
    vf 3     link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff, spoof checking on, link-state auto, trust off
    vf 4     link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff, spoof checking on, link-state auto, trust off
    vf 5     link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff, spoof checking on, link-state auto, trust off
    altname enp81s0f1
```

Modify configMap as following:

```
vi ~/go/src/github.com/intel/sriov-network-device-plugin/deployments
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: sriovdp-config
  namespace: kube-system
data:
  config.json: |
    {
        "resourceList": [{
                "resourceName": "intel_sriov_odu",
                "selectors": {
                    "vendors": ["8086"],
                    "devices": ["1889"],
                    "drivers": ["enp81s0f0"]
                }
            },
            {
                "resourceName": "intel_sriov_oru",
                "selectors": {
                    "vendors": ["8086"],
                    "devices": ["1889"],
                    "drivers": ["vfio-pci"],
                    "pfNames": ["enp81s0f1"]
                }
            },
            {
                "resourceName": "intel_fec_5g",
                 "deviceType": "accelerator",
                 "selectors": {
                     "vendors": ["8086"],
                     "devices": ["0d5d"]
                }
            },
            {
                "resourceName": "mlnx_sriov_rdma",
                "selectors": {
                    "vendors": ["15b3"],
                    "devices": ["1017"],
                    "drivers": ["mlx5_core"],
                    "isRdma": true
                }
            }
        ]
    }
~
```

```
kubectl create -f configMap.yaml
```

Modify /var/flexran/build/docker/flexran_testmac_mode.yaml according to your specs:

```
kubectl create -f flexran_testmac_mode.yaml
```

```
XR12-B:/var/flexran/build/docker # kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
flexran-binary-release    2/2     Running   0          110s
testpod1                  1/1     Running   0          16h
```

5.5 Testing FlexRAN™ Timer Mode in Containers

To demonstrate simple functionality:

In the 1st terminal run:

```
kubectl exec –it flexran-binary-release –c flexran-l1app – bash
Start l1.sh -e
```

In the 2nd terminal run:

```
kubectl exec –it flexran-binary-release –c flexran-testmac -- bash
```

Note

Make sure that your dpdk directory mapped in the yaml file.

```
 Other tests such as xRAN Mode and a Helm Chart test can be run as well as described in
section 5.2 and 5.3 of Intel document 575834-15.0 [Installation Guide Software Release
 v22.07]
```

As a simplified solution, a pre-configured Intel® FlexRAN™ helm chart as well as all required CNI plugins, can be posted on Rancher Marketplace to simplify deployment at a large scale.

6 Summary

Building, testing, and deploying a properly configured Intel® FlexRAN™ implementation can show the benefits of VNFs and vRAN with Intel® Xeon® Scalable Processors and Intel® Advanced Vector Extensions.

SUSE provides all the elements for an open-source, enterprise-grade, software-defined stack for cloud-native orchestration and management. SUSE Linux Enterprise (with Real Time extensions), SUSE Linux Enterprise Micro Real Time, Rancher Kubernetes Engine v2 (RKE2) and Rancher Management were used and illustrated as key ingredients to simplify the deployment of Intel® FlexRAN™.

7 Reference

- https://github.com/intel/FlexRAN
- https://www.intel.com/content/www/us/en/developer/videos/how-radio-access-network-is-being-virtualized-and-the-role-of-flexran.html?wapkw=FlexRan
- https://www.intel.com/content/www/us/en/developer/topic-technology/edge-5g/tools/flexran.html?wapkw=FlexRan
- https://www.intel.com/content/www/us/en/communications/5g-get-your-infrastructure-ready-guide.html
- https://docs.rke2.io/install/quickstart/
- https://documentation.suse.com/trd/kubernetes/pdf/kubernetes_ri_rancher-k3s-slemicro_color_en.pdf
- https://documentation.suse.com/sle-rt/15-SP3/
- https://documentation.suse.com/sles/15-SP3/pdf/book-container_color_en.pdf
- https://documentation.suse.com/sle-micro/5.3/

8 Legal notice

9 GNU Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all

Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/ ↗.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
   Permission is granted to copy, distribute and/or modify this document
   under the terms of the GNU Free Documentation License, Version 1.2
```

```
   or any later version published by the Free Software Foundation;
   with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
   A copy of the license is included in the section entitled “GNU
   Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “ with… Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
   Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.