[OAI] Installation Guide(uncompression)

support FHI7.2 / on the Server with RHEL

Branch: use_msgq

留言

🎯 **Readme:**

- Corresponding Titles and Numbers are available in **OAI Tutorial**
- Troubleshooting note has been compiled and placed at the end of this document.
- Partitioning script has been generated.

Contributer:

- `Ming`
- `Yueh-Huan`

⚓ **Installation environment:**

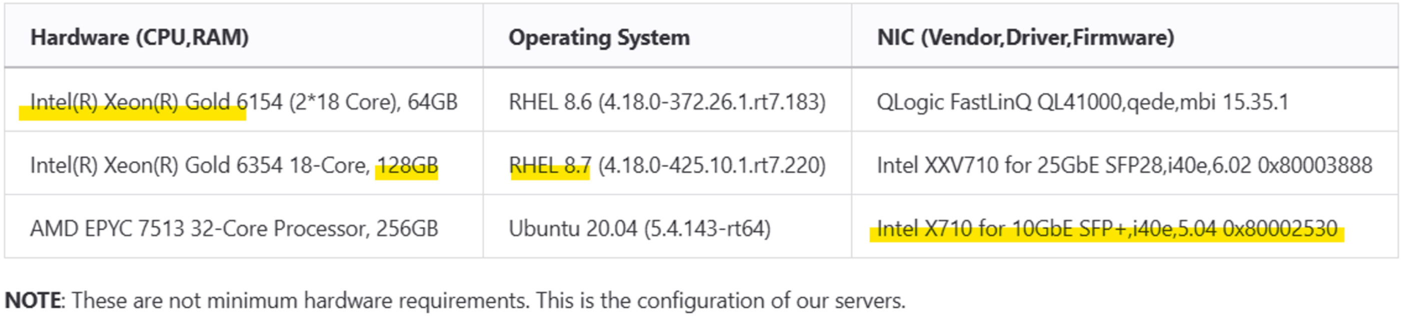
- **Prerequisites**

> Dell R740

Hardware (CPU, RAM, Disk)	Operating System	NIC (Vendor,Driver,Firmware)
Intel® Xeon® Gold 6226R x2, 128GB, 600GB	RHEL 8.7	Intel X710 for 10GbE SFP+,i40e,5.04 0x80002530

> OAI Branch: use_msgq

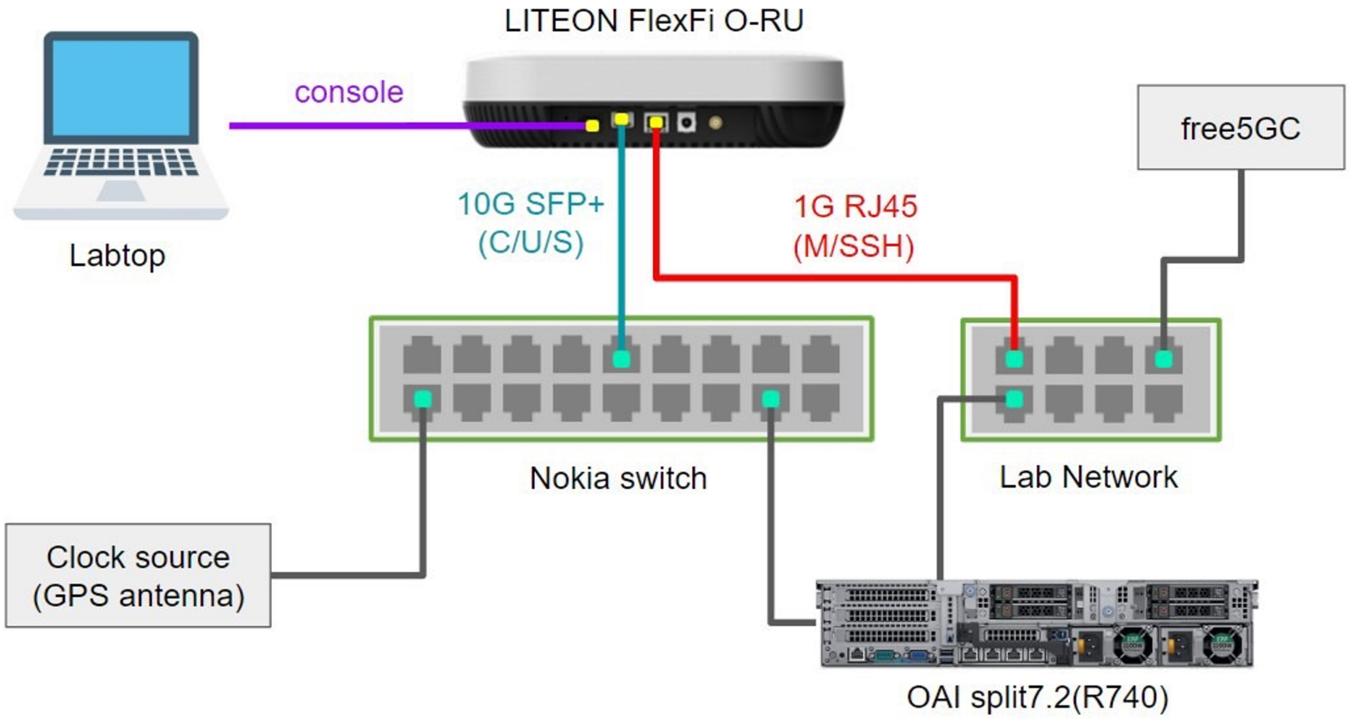
- **The hardware on which OAI have tried**



Hardware (CPU,RAM)	Operating System	NIC (Vendor,Driver,Firmware)
Intel(R) Xeon(R) Gold 6154 (2*18 Core), 64GB	RHEL 8.6 (4.18.0-372.26.1.rt7.183)	QLogic FastLinQ QL41000,qede,mbi 15.35.1
Intel(R) Xeon(R) Gold 6354 18-Core, 128GB	RHEL 8.7 (4.18.0-425.10.1.rt7.220)	Intel XXV710 for 25GbE SFP28,i40e,6.02 0x80003888
AMD EPYC 7513 32-Core Processor, 256GB	Ubuntu 20.04 (5.4.143-rt64)	Intel X710 for 10GbE SFP+,i40e,5.04 0x80002530

NOTE: These are not minimum hardware requirements. This is the configuration of our servers.

- **Network Topology**



Reference:

- **ORAN_FHI7.2_Tutorial**
- LITEON build Note
- LITEON install OAI gNB LOG

1. Introduction

Put our harware info, expected software info here.

Check your software info

Script file: SHOW_INFO.sh

- **System Information**

	info
Red Hat Release	Red Hat Enterprise Linux 8.8 (Ootpa)
Kernel Information	Linux R740 4.18.0-425.10.1.rt7.220.el8_7.x86_64
CPU Information	Intel® Xeon® Gold 6226R CPU @ 2.90GHz
PCI Devices (X710)	3b:00.0 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 02)
	3b:00.1 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 02)

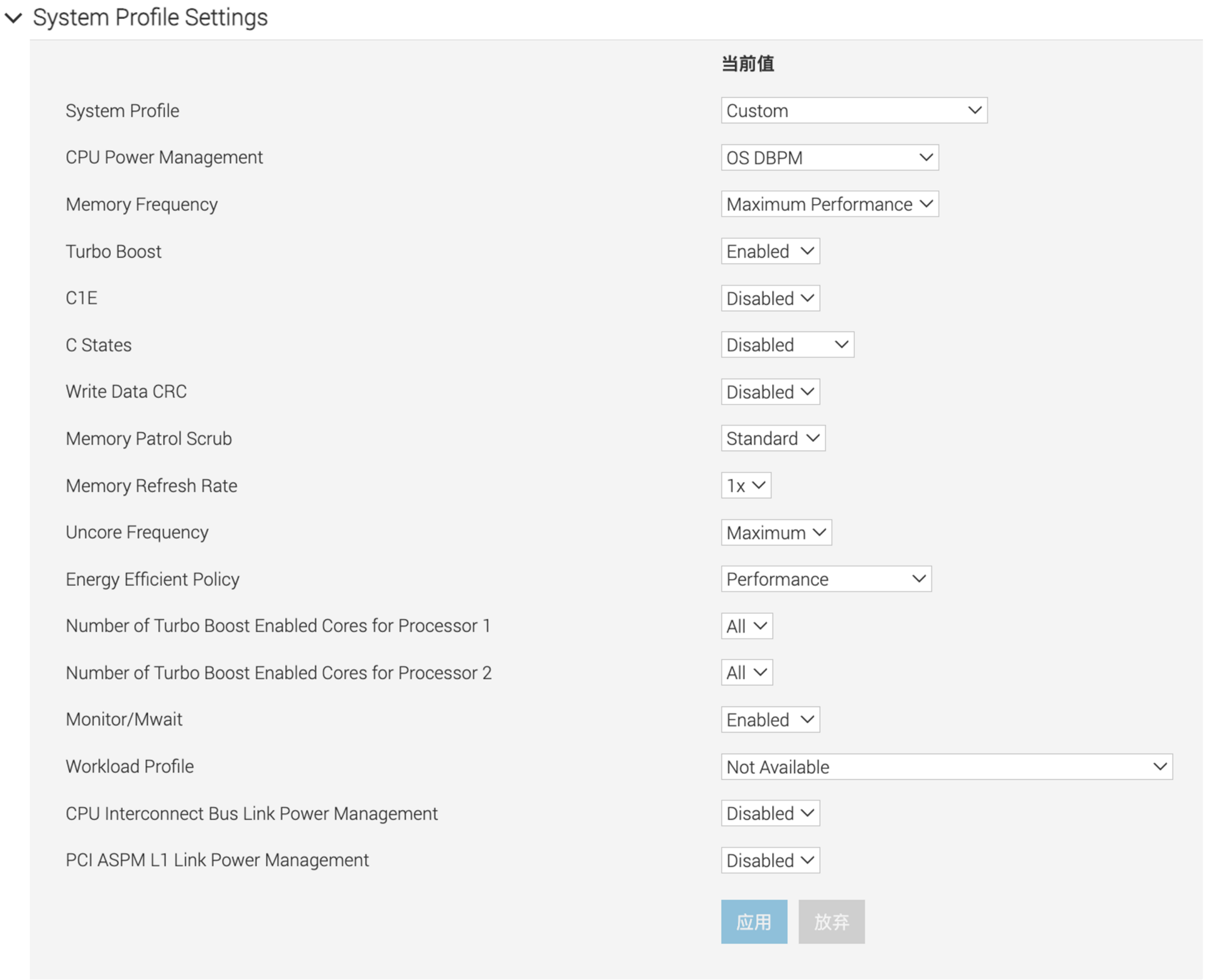
2. BIOS Setting

2.1 Enable Customize CPU Frequency

Due to the requirement of OAI, we need to enable this function. Sometimes, it will be blocked

Here, we are using Dell R740 as an example.

1. Configure BIOS



System Profile Settings

	当前值
System Profile	Custom
CPU Power Management	OS DBPM
Memory Frequency	Maximum Performance
Turbo Boost	Enabled
C1E	Disabled
C States	Disabled
Write Data CRC	Disabled
Memory Patrol Scrub	Standard
Memory Refresh Rate	1x
Uncore Frequency	Maximum
Energy Efficient Policy	Performance
Number of Turbo Boost Enabled Cores for Processor 1	All
Number of Turbo Boost Enabled Cores for Processor 2	All
Monitor/Mwait	Enabled
Workload Profile	Not Available
CPU Interconnect Bus Link Power Management	Disabled
PCI ASPM L1 Link Power Management	Disabled

2. Reboot Server

3. Check `cpupower` :It will display the available frequency steps

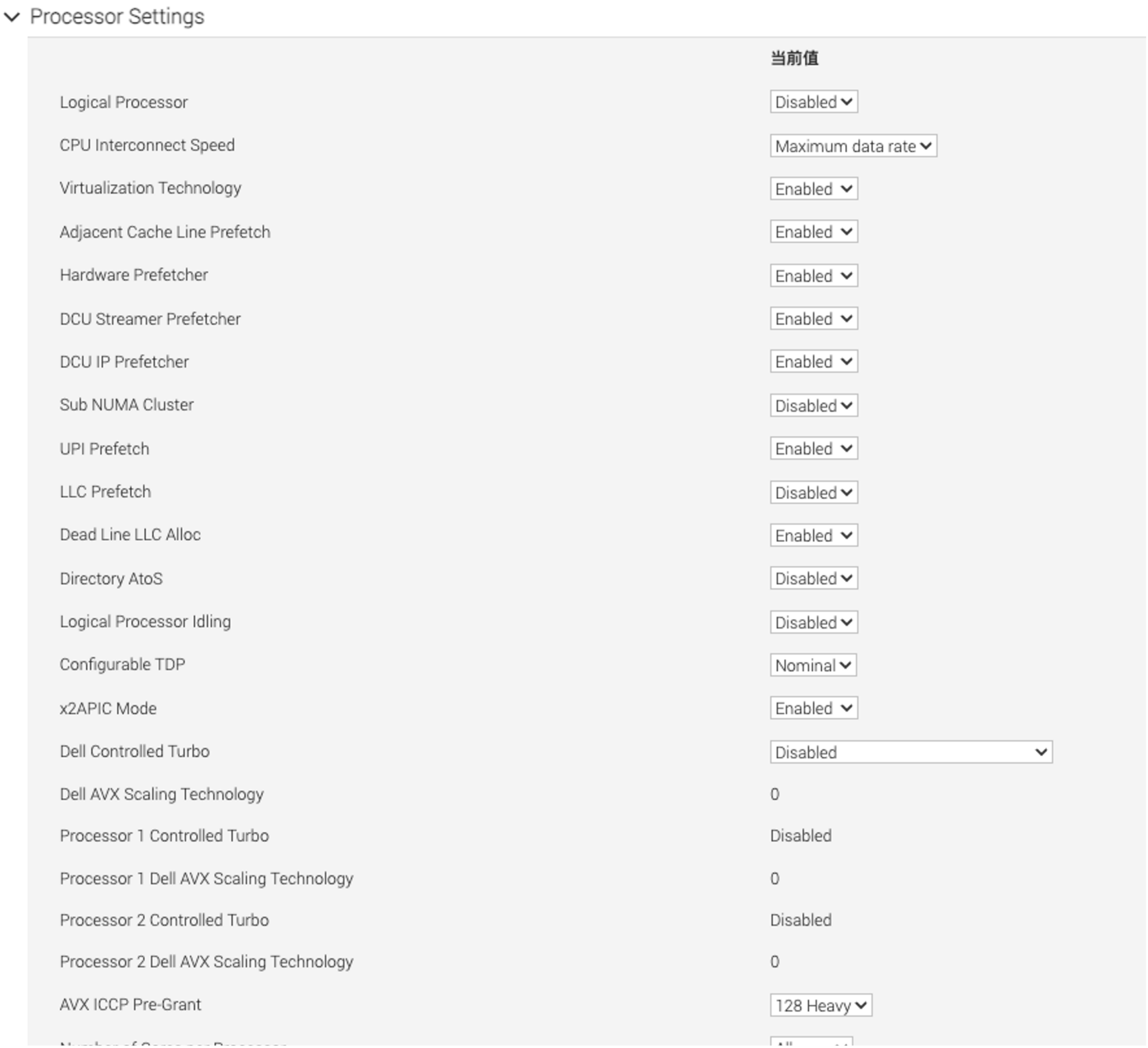
```
# cpupower frequency-info analyzing CPU 0: driver: acpi-cpufreq CPUs which run at the same hardware frequency: 0 CPUs which need to have their frequency coordinated by software: 0 maximum transition latency: 10.0 us hardware limits: 1.20 GHz - 2.90 GHz available frequency steps: 2.90 GHz, 2.90 GHz, 2.80 GHz, 2.70 GHz, 2.50 GHz, 2.40 GHz, 2.30 GHz, 2.20 GHz, 2.00 GHz, 1.90 GHz, 1.80 GHz, 1.70 GHz, 1.60 GHz, 1.40 GHz, 1.30 GHz, 1.20 GHz available cpufreq governors: conservative ondemand userspace powersave performance schedutil current policy: frequency should be within 1.20 GHz and 2.90 GHz. The governor "performance" may decide which speed to use within this range. current CPU frequency: Unable to call hardware current CPU frequency: 2.90 GHz (asserted by call to kernel) boost state support: Supported: yes Active: yes
```

1. Check `CPU file` If we enable custom CPU frequency successfully, it can display something like this. Otherwise, you can not find this file.

```
# cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq 1200000
```

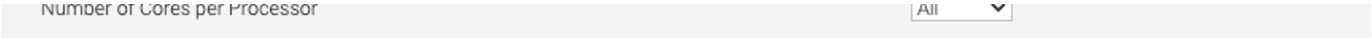
2.2 Disable Hyper Threading

Refer to the setting of OAI, we disable the hyper threading(Logical Processor).



Processor Settings

	当前值
Logical Processor	Disabled
CPU Interconnect Speed	Maximum data rate
Virtualization Technology	Enabled
Adjacent Cache Line Prefetch	Enabled
Hardware Prefetcher	Enabled
DCU Streamer Prefetcher	Enabled
DCU IP Prefetcher	Enabled
Sub NUMA Cluster	Disabled
UPI Prefetch	Enabled
LLC Prefetch	Disabled
Dead Line LLC Alloc	Enabled
Directory AtoS	Disabled
Logical Processor Idling	Disabled
Configurable TDP	Nominal
x2APIC Mode	Enabled
Dell Controlled Turbo	Disabled
Dell AVX Scaling Technology	0
Processor 1 Controlled Turbo	Disabled
Processor 1 Dell AVX Scaling Technology	0
Processor 2 Controlled Turbo	Disabled
Processor 2 Dell AVX Scaling Technology	0
AVX ICCP Pre-Grant	128 Heavy
Number of Cores per Processor	All



Number of Cores per Processor All

3. System, Tools and Dependency

Before installing the software package, you need to subscribe the package first.

Follow the Note to Subscribe

Register Red Hat Account

1. Go to https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux to sign up for a Red Hat account.
2. Fill in the required information like name, email, company etc to complete the registration.
3. After registration, login to https://access.redhat.com/ with the new account.
4. Under “Subscriptions”, click “Get a Red Hat Subscription” to get a free Red Hat Developer Subscription.
5. Follow the prompts to activate the free subscription. Make sure to note down the username and password.
6. The free Developer Subscription allows you to access RHEL for development usage.
7. Apply for Email Verification

Attach Subscription to RHEL OS

1. Open terminal, use `subscription-manager register` to register.
2. Enter Red Hat account and password when prompted.
3. After successful registration, use `subscription-manager list` to view available subscriptions.

- Command

```
subscription-manager register --username <YOUR_USERNAME> --password <YOUR_PASSWO
RD> --auto-attach
```

3.1 Using Real-Time Kernel

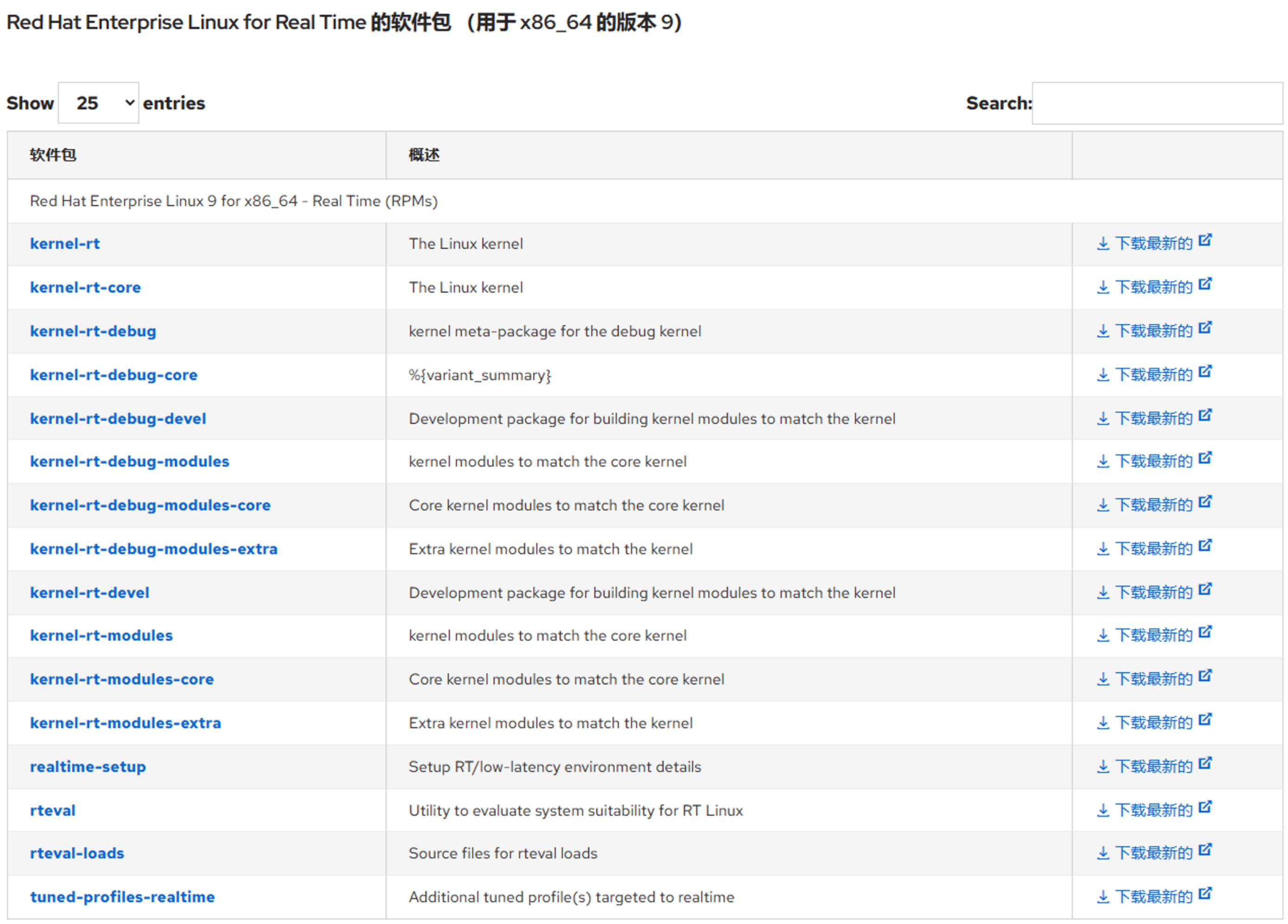
Change kernel

Here, we are using `kernel-rt-5.14` as an example.

Reference doc: RedHat 教學更換Kernel

Reference download (RH8_kernel).rpm

Reference download (RH9_kernel).rpm



Red Hat Enterprise Linux for Real Time 的软件包 （用于 x86_64 的版本 9）

Show 25 entries Search:

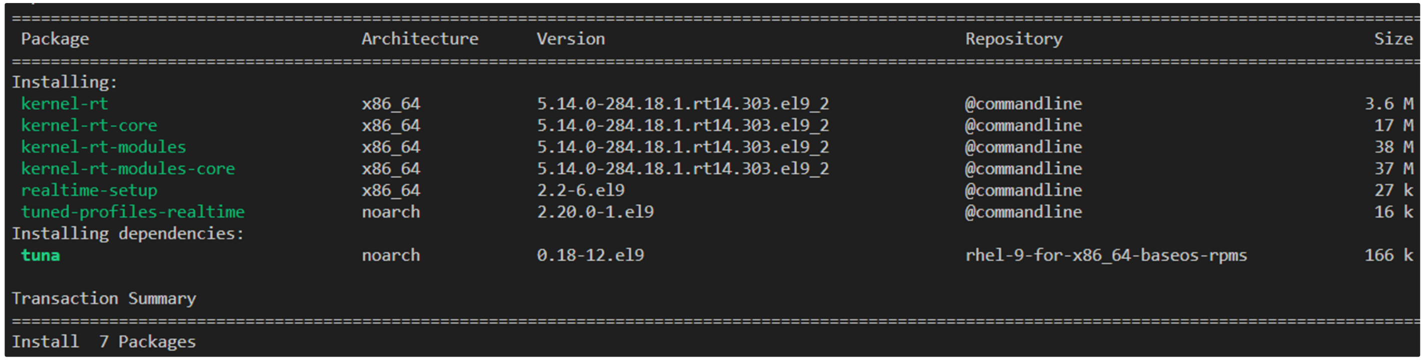
软件包	概述	
Red Hat Enterprise Linux 9 for x86_64 - Real Time (RPMs)		
kernel-rt	The Linux kernel	下载最新的
kernel-rt-core	The Linux kernel	下载最新的
kernel-rt-debug	kernel meta-package for the debug kernel	下载最新的
kernel-rt-debug-core	%{variant_summary}	下载最新的
kernel-rt-debug-devel	Development package for building kernel modules to match the kernel	下载最新的
kernel-rt-debug-modules	kernel modules to match the core kernel	下载最新的
kernel-rt-debug-modules-core	Core kernel modules to match the core kernel	下载最新的
kernel-rt-debug-modules-extra	Extra kernel modules to match the kernel	下载最新的
kernel-rt-devel	Development package for building kernel modules to match the kernel	下载最新的
kernel-rt-modules	kernel modules to match the core kernel	下载最新的
kernel-rt-modules-core	Core kernel modules to match the core kernel	下载最新的
kernel-rt-modules-extra	Extra kernel modules to match the kernel	下载最新的
realtime-setup	Setup RT/low-latency environment details	下载最新的
rteval	Utility to evaluate system suitability for RT Linux	下载最新的
rteval-loads	Source files for rteval loads	下载最新的
tuned-profiles-realtime	Additional tuned profile(s) targeted to realtime	下载最新的

> You need to have an account and be logged in to download.

Name (.rpm)	Rel.
kernel-rt	5.14.0-284.18.1.rt14.303.el9_2
realtime-setup	2.2-6.el9
tuned-profiles-realtime	2.20.0-1.el9
kernel-rt-core	5.14.0-284.18.1.rt14.303.el9_2
kernel-rt-modules	5.14.0-284.18.1.rt14.303.el9_2
kernel-rt-modules-core	5.14.0-284.18.1.rt14.303.el9_2

Also you can download from my OneDrive

```
sudo yum install ./kernel-rt-5.14.0-284.18.1.rt14.303.el9_2.x86_64.rpm ./realtime-setup-2.2-6.el9.x86_64.rpm ./tuned-profiles-realtime-2.20.0-1.el9.noarch.rpm ./kernel-rt-core-5.14.0-284.18.1.rt14.303.el9_2.x86_64.rpm ./kernel-rt-modules-5.14.0-284.18.1.rt14.303.el9_2.x86_64.rpm ./kernel-rt-modules-core-5.14.0-284.18.1.rt14.303.el9_2.x86_64.rpm
```



```
================================================================================================================================================
 Package                            Architecture      Version                                     Repository                              Size
================================================================================================================================================
Installing:
 kernel-rt                          x86_64            5.14.0-284.18.1.rt14.303.el9_2              @commandline                           3.6 M
 kernel-rt-core                     x86_64            5.14.0-284.18.1.rt14.303.el9_2              @commandline                            17 M
 kernel-rt-modules                  x86_64            5.14.0-284.18.1.rt14.303.el9_2              @commandline                            38 M
 kernel-rt-modules-core             x86_64            5.14.0-284.18.1.rt14.303.el9_2              @commandline                            37 M
 realtime-setup                     x86_64            2.2-6.el9                                   @commandline                            27 k
 tuned-profiles-realtime            noarch            2.20.0-1.el9                                @commandline                            16 k
Installing dependencies:
 tuna                               noarch            0.18-12.el9                                 rhel-9-for-x86_64-baseos-rpms          166 k

Transaction Summary
================================================================================================================================================
Install  7 Packages
```

To view the default kernel:

```
sudo grubby --default-kernel
```

> If not you want kernel, you need to change default kernel

```
sudo grubby --set-default=/boot/vmlinuz-<YOUR_NEW_KERNEL> sudo sed -i 's/UPDATEDEFAULT=.*/UPDATEDEFAULT=yes/g' /etc/sysconfig/kernel sudo sed -i 's/DEFAULTKERNEL=.*/DEFAULTKERNEL=kernel-rt-core/g' /etc/sysconfig/kernel # Reboot than check it! sudo reboot uname -r
```

3.2 Setup GRUB

Srcipt file: Modifly_GRUB.sh

Modify grub file

add configuration from OAI Tutorial: `igb.max_vfs=2 intel_iommu=on iommu=pt intel_pstate=disable nosoftlockup tsc=nowatchdog mitigations=off cgroup_memory=1 cgroup_enable=memory mce=off idle=poll hugepagesz=1G hugepages=40 hugepagesz=2M hugepages=0 default_hugepagesz=1G selinux=0 enforcing=0 nmi_watchdog=0 softlockup_panic=0 audit=0 skew_tick=1 isolcpus=managed_irq,domain,0-2,8-17 nohz_full=0-2,8-17 rcu_nocbs=0-2,8-17 rcu_nocb_poll`

Follow anthor Tutorial `If you have a server with 2 NUMA nodes, you should run DPDK and the ru_thread on the first NUMA node, and OAI on the other.` You need to check your NUMA CPU(s) setting.

Check your NUMA CPU(s) set

```
lscpu | grep -e "NUMA node"
```

```
NUMA node(s): 2 NUMA node0 CPU(s): 0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30 NU
MA node1 CPU(s): 1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31
```

Replace `0-2,8-17` with `0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30`.

```
sudo vi /etc/default/grub
```

From

```
GRUB_CMDLINE_LINUX="crashkernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet"
```

Change into

```
GRUB_CMDLINE_LINUX="pci=assign-busses pci=realloc igb.max_vfs=2 intel_iommu=on iommu=pt intel_pstate=disable nosoftlockup tsc=nowatchdog mitigations=off cgroup_memory=1 cgroup_enable=memory mce=off idle=poll hugepagesz=1G hugepages=40 hugepagesz=2M hugepages=0 default_hugepagesz=1G selinux=0 enforcing=0 audit=0 nmi_watchdog=0 softlockup_panic=0 skew_tick=1 isolcpus=managed_irq,domain,0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30 nohz_full=0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30 rcu_nocbs=0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30 rcu_nocb_poll"
```

GRUB configuration updated

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

3.3 Setup SCTP

Reference:

- https://sysadmin.lk/setup-sctp-in-red-hat-centos-linux-8/
- https://access.redhat.com/solutions/6625041

```
yum install kernel-rt-modules-extra.x86_64 yum install kernel-modules-extra
```

```
yum install -y lksctp-tools lksctp-tools-devel lksctp-tools-doc
```

- add text `sctp` in `/etc/modules-load.d/sctp.conf`

```
vi /etc/modules-load.d/sctp.conf
```

Commenting out the SCTP blacklist:

In the file `/etc/modprobe.d/sctp-blacklist.conf`, locate the line “blacklist sctp” and add a hash symbol “#” at the beginning of the line to comment it out. This action will remove the blacklist restriction on SCTP, allowing the system to automatically load the SCTP module.

```
vi /etc/modprobe.d/sctp-blacklist.conf
```

```
# blacklist sctp
```

- Similarly, perform the same action as the previous step in `/etc/modprobe.d/sctp_diag-blacklist.conf`.

Enabling:

- Rebooting: `sudo reboot`
- Manually starting SCTP: `modprobe sctp`

Check some command such as ncat provided from the nmap-ncat package to ensure sctp sockets can be created

```
lsmod | grep sctp ncat --sctp -k -l 127.0.0.1 8192 ## SHOW LOGS sctp 421888 4 ip
6_udp_tunnel 16384 1 sctp udp_tunnel 20480 1 sctp libcrc32c 16384 5 nf_conntrac
k,nf_nat,nf_tables,xfs,sctp ## In another terminal, check ss ss -pneomSa | grep
-A 1 8192 ## In another terminal, SHOW LOGS [oai@R740 ~]$ ss -pneomSa | grep -A
1 8192 LISTEN 0 10 127.0.0.1:8192 0.0.0.0:* users:(("ncat",pid=53414,fd=3)) uid:
1000 ino:265056 sk:1 <-> skmem:(r0,rb212992,t0,tb212992,f0,w0,o0,bl0,d0) locals:
127.0.0.1 checksctp ## SHOW LOG => SCTP supported
```

3.4 Install Tools and Dependency

Script file: Install.sh

```
# set the server to maximum performance mode. sudo tuned-adm profile realtime
# Updating RedHat System sudo yum update # yum install sudo yum install python3-pip gcc gcc-c++ kernel-devel make linuxptp # pip install sudo pip3 install meson==0.58.2 # localinstall wget https://vault.centos.org/centos/8/PowerTools/x86_64/os/Packages/ninja-build-1.8.2-1.el8.x86_64.rpm sudo yum -y localinstall ninja-build-1.8.2-1.el8.x86_64.rpm # Others pmc help -y
```

3.5 DPDK(Data Plane Development Kit)

```
wget http://fast.dpdk.org/rel/dpdk-20.05.tar.xz tar xvf dpdk-20.05.tar.xz cd dpdk-20.05 meson build cd build sudo ninja sudo ninja install yes | sudo dnf install numactl yes | sudo dnf install numactl-devel cd .. make install T=x86_64-native-linuxapp-gcc
```

3.6 Linux PTP configuration

Srcipt file: LinuxPTP.sh

```
git clone http://git.code.sf.net/p/linuxptp/code linuxptp cd ~/linuxptp/ git checkout v2.0 sudo make install
```

If there is no default-1.cfg change in `default.cfg`

- **domainNumber:** `24`

- **dataset_comparison:** `G.8275.x`
- **network_transport:** `L2`
- **time_stamping:** `hardware`

```
vi configs/default.cfg cat configs/default.cfg
```

Check

```
[global] # # Default Data Set # twoStepFlag 1 slaveOnly 0 priority1 128 priority
2 128 domainNumber 24 #utc_offset 37 clockClass 248 clockAccuracy 0xFE offsetSca
ledLogVariance 0xFFFF free_running 0 freq_est_interval 1 dscp_event 0 dscp_gener
al 0 dataset_comparison G.8275.x G.8275.defaultDS.localPriority 128 ...
```

4. Build OAI-FHI gNB

Script file: Build_OAI-FHIgNB.sh

4.1 Git clone OAI and edit file

Git clone

```
## Download openairinterface and checkout use_msgq branch git clone https://gitlab.eurecom.fr/oai/openairinterface5g.git cd ~/openairinterface5g/ git checkout use_msgq
```

Change from `"rhel8.7") return 0 ;;` into `"rhel8.8") return 0 ;;`

```
cd ~/openairinterface5g/cmake_targets/tools vim build_helper
```

4.2 Build ORAN Fronthaul Interface Library

```
cd ~ git clone https://gerrit.o-ran-sc.org/r/o-du/phy.git cd ~/phy git checkout oran_release_bronze_v1.1 # Copy openairinterface fils and update patch cp ~/openairinterface5g/cmake_targets/tools/oran_fhi_integration_patches/oran-fhi-* . git apply oran-fhi-1-compile-libxran-using-gcc-and-disable-avx512.patch git apply oran-fhi-2-return-correct-slot_id.patch git apply oran-fhi-3-disable-pkt-validate-at-process_mbuf.patch git apply oran-fhi-4-process_all_rx_ring.patch git apply oran-fhi-5-remove-not-used-dependencies.patch # Set Gobal export XRAN_LIB_DIR=~/phy/fhi_lib/lib/build export XRAN_DIR=~/phy/fhi_lib export RTE_SDK=~/dpdk-20.05 export RTE_TARGET=x86_64-native-linuxapp-gcc export RTE_INCLUDE=${RTE_SDK}/${RTE_TARGET}/include # Build OAI cd ~/phy/fhi_lib/ ./build.sh
```

4.3 Build OAI gNB

Script file: Build_OAI.sh

Install lack lib from build rpm

```
yum install -y wget wget https://vault.centos.org/centos/8/PowerTools/x86_64/os/Packages/guile-devel-2.0.14-7.el8.x86_64.rpm yum -y groupinstall "Development Tools" wget https://vault.centos.org/centos/8/PowerTools/x86_64/os/Packages/gc-devel-7.6.4-3.el8.x86_64.rpm yum -y localinstall gc-devel-7.6.4-3.el8.x86_64.rpm yum -y localinstall guile-devel-2.0.14-7.el8.x86_64.rpm wget https://vault.centos.org/centos/8/PowerTools/x86_64/os/Packages/libconfig-devel-1.5-9.el8.x86_64.rpm yum -y localinstall libconfig-devel-1.5-9.el8.x86_64.rpm wget https://vault.centos.org/centos/8/PowerTools/x86_64/os/Packages/lapack-devel-3.8.0-8.el8.x86_64.rpm wget https://vault.centos.org/centos/8/BaseOS/x86_64/os/Packages/pkgconf-pkg-config-1.4.2-1.el8.x86_64.rpm yum -y localinstall pkgconf-pkg-config-1.4.2-1.el8.x86_64.rpm wget https://vault.centos.org/centos/8/PowerTools/x86_64/os/Packages/blas-devel-3.8.0-8.el8.x86_64.rpm yum -y localinstall blas-devel-3.8.0-8.el8.x86_64.rpm yum -y localinstall lapack-devel-3.8.0-8.el8.x86_64.rpm yum -y install python27
```

> if oran_fhlib_5g build fail, add -fPIC to CFLAGS in all makefiles in dpdk, then build dpdk again

List of modified files in Makefile and notes on what was changed：

https://hackmd.io/@MingHung/EditMakefile

```
# Granting Execute Permission to a File Using chmod Command chmod +x /home/oai/dpdk-20.05/buildtools/map-list-symbol.sh chmod +x /home/oai/dpdk-20.05/buildtools/check-symbols.sh chmod +x /home/oai/dpdk-20.05/buildtools/gen-config-h.sh # Re-build DPDK for OAI cd ~/dpdk-20.05 meson setup --wipe ~/dpdk-20.05/build meson build cd build sudo ninja sudo ninja install cd .. make config T=x86_64-native-linuxapp-gcc make install T=x86_64-native-linuxapp-gcc
```

Opening a new terminal requires reconfiguring the parameters once.

```
export XRAN_LIB_DIR=~/phy/fhi_lib/lib/build export XRAN_DIR=~/phy/fhi_lib export RTE_SDK=~/dpdk-20.05 export RTE_TARGET=x86_64-native-linuxapp-gcc export RTE_INCLUDE=${RTE_SDK}/${RTE_TARGET}/include
```

Build OAI

```
cd ~/openairinterface5g/cmake_targets/ ./build_oai --gNB --ninja -t oran_fhlib_5g -I
```

Build OAI success log

- https://hackmd.io/@MingHung/build_oai

5. Configuration

5.1 Find the Parameters

```
* DU Mac-address: Parameter o_du_macaddr * RU MAC Address: Parameter o_ru_macadd
r * PCI address: Parameter dpdk_dev_up and dpdk_dev_cp
```

- Interface for C/U Plane
- Interface for S Plane (PTP)

5.2 Configure O-RAN Fronthaul Interface C/U Plane

Intro:

This configuration file is used for FHI library.

Let’s copy the configuration from default setting.

```
cd ~/openairinterface5g/cmake_targets/ran_build/build cp ../../../targets/PROJEC
TS/GENERIC-NR-5GC/CONF/oran.fhi.json ./conf.json
```

- **Reference:**
  - `openairinterface5g/radio/ETHERNET/oran/5g/xran_lib_wrap.hpp`
  - `openairinterface5g/radio/ETHERNET/oran/5g/xran_lib_wrap.cpp`

Edit the configuration:

```
vim ~/openairinterface5g/cmake_targets/ran_build/build/conf.json
```

We need to configure the parameter for `4 ~ 5`, `17 ~ 20` **based on your system.**

```
"GLOBAL": { "io_cfg": { "dpdk_dev_up": "0000:3b:0a.1", "dpdk_dev_cp": "0000:3b:0a.0", "bbdev_mode": "none", "system_core": 0, "core": 1, "pkt_proc_core": 2, "timing_core": 1, "pkt_aux_core": 1, "txq_id": 0, "rxq_id": 0, "dpdkBasebandFecMode": 0, "dpdkBasebandDevice": "", "mtu": 8870, "o_du_macaddr": "00:11:22:33:44:66", "o_ru_macaddr": "00:aa:ff:bb:ff:cc", "cp_vlan_tag": 3, "up_vlan_tag": 3 },
```

Parameter	Meaning	Value
`dpdk_dev_up`	PCI Bus address used for fronthaul user-plane	`0000:3b:0a.1`
`dpdk_dev_cp`	PCI Bus address used for fronthaul control-plane	`0000:3b:0a.0`
`o_du_macaddr`	MAC address for O-DU frontfaul NIC	`00:11:22:33:44:66`
`o_ru_macaddr`	MAC address for O-RU frontfaul NIC	`00:aa:ff:bb:ff:cc`
`cp_vlan_tag`	vlan id for fronthaul control-plane	`3`
`up_vlan_tag`	vlan id for fronthaul user-plane	`3`

5.3 Configure O-RAN Fronthaul Interface S Plane

Reference: ORAN_FHI7.2_Tutorial.md

You can refer to the following O-RAN link for PTP information.

First, disable `ntp` and/or `chrony`; they cause problems with `phc2sys`.

In our setup we used Fibrolan Falcon-RX for PTP grandmaster. Unlike the O-RAN tutorial, we install `ptp4l` (v3.1.1) using the package manager, and use a simple configuration file as shown below, where the PTP grandmaster is reachable via interafce `ens7f1`.

```
$ cat /tmp/ptp4l.conf [global] domainNumber 24 slaveOnly 1 verbose 1 network_transport L2 time_stamping hardware tx_timestamp_timeout 100 [ens7f1] $ sudo ptp4l -i ens7f1 -m -H -2 -s -f /etc/ptp4l.conf # note: ptp4l.service should not run yet $ sudo phc2sys -w -m -s ens7f1 -R 8 -f /etc/ptp4l.conf # same as above
```

Note to the above: this tutorial assumes a machine with a Intel E-810 NICE (using `ice` driver), so we needed to set `tx_timestamp_timeout` to a high value. In other cases, `1` is enough.

If the offset is high, make sure you are using `skew_tick=1` in your kernel commandline.

If everything works, enable the ptp4l system daemon (note: daemon options are in `/etc/sysconfig/ptp4l`) and reboot:

```
$ sudo systemctl start ptp4l.service $ sudo systemctl status ptp4l # should be fine explain more what $ cat /etc/sysconfig/phc2sys OPTIONS="-a -r -r -n 24" # man! $ sudo systemctl start phc2sys.service
```

@sagar: Pay attention that the `freq` parameters are low

5.4 Configure OAI gNB

Path: `/home/oai/openairinterface5g/targets/PROJECTS/GENERIC-NR-5GC/CONF/oran.fh.band78.fr1.273PRB.conf`

```
NETWORK_INTERFACES : { GNB_INTERFACE_NAME_FOR_NG_AMF = "eno1np0"; GNB_IPV4_ADDRESS_FOR_NG_AMF = "192.168.8.29/24"; GNB_INTERFACE_NAME_FOR_NGU = "eno1np0"; GNB_IPV4_ADDRESS_FOR_NGU = "192.168.8.29/24"; GNB_PORT_FOR_S1U = 2152; # Spec 2152 };
```

> eno1np0: Used for connecting to the laboratory network.

---

6. Execution

6.1 Create Virtual Function

```
sudo su # initial VF echo 0 > /sys/class/net/ens1f1/device/sriov_numvfs echo 2 > /sys/class/net/ens1f1/device/sriov_numvfs # configure virtual function ip link set ens1f1 vf 0 mac 00:11:22:33:44:66 vlan 3 spoofchk off ip link set ens1f1 vf 1 mac 00:11:22:33:44:66 vlan 3 spoofchk off exit
```

```
# Check configuration ip link show ens1f1
```

```
7: ens1f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEF
AULT group default qlen 1000 link/ether 40:a6:b7:92:c4:75 brd ff:ff:ff:ff:ff:ff
vf 0 link/ether 00:11:22:33:44:66 brd ff:ff:ff:ff:ff:ff, vlan 3, spoof checking
off, link-state auto, trust off vf 1 link/ether 00:11:22:33:44:66 brd ff:ff:ff:f
f:ff:ff, vlan 3, spoof checking off, link-state auto, trust off altname enp59s0f
1
```

6.2 Bind DPDK

Intro:

Here are the vitural functions we want to bind.

VF Name	PCI Address
ens1f1v0	0000:3b:0a.0
ens1f1v1	0000:3b:0a.1

Bind:

```
sudo su # unbind python3 /usr/local/bin/dpdk-devbind.py --force --unbind 0000:3
b:0a.0 0000:3b:0a.1 # reload module modprobe vfio_pci # bind python3 /usr/local/
bin/dpdk-devbind.py --bind vfio-pci 0000:3b:0a.0 0000:3b:0a.1 exit
```

Check:

```
sudo python3 /usr/local/bin/dpdk-devbind.py --status
```

```
Network devices using DPDK-compatible driver ===================================
========= 0000:3b:0a.0 'Ethernet Virtual Function 700 Series 154c' drv=vfio-pci
unused=iavf 0000:3b:0a.1 'Ethernet Virtual Function 700 Series 154c' drv=vfio-pc
i unused=iavf Network devices using kernel driver ==============================
===== 0000:01:00.0 'NetXtreme BCM5720 Gigabit Ethernet PCIe 165f' if=eno3 drv=tg
3 unused=vfio-pci 0000:01:00.1 'NetXtreme BCM5720 Gigabit Ethernet PCIe 165f' if
=eno4 drv=tg3 unused=vfio-pci 0000:19:00.0 'BCM57416 NetXtreme-E Dual-Media 10G
RDMA Ethernet Controller 16d8' if=eno1np0 drv=bnxt_en unused=vfio-pci *Active* 0
000:19:00.1 'BCM57416 NetXtreme-E Dual-Media 10G RDMA Ethernet Controller 16d8'
if=eno2np1 drv=bnxt_en unused=vfio-pci 0000:3b:00.0 'Ethernet Controller X710 fo
r 10GbE SFP+ 1572' if=ens1f0 drv=i40e unused=vfio-pci 0000:3b:00.1 'Ethernet Con
troller X710 for 10GbE SFP+ 1572' if=ens1f1 drv=i40e unused=vfio-pci
```

6.3 Sync up using PTP

Execution:

```
cd ~/linuxptp sudo ptp4l -i ens1f1 -m -H -2 -s -f configs/default.cfg > ptp4l.lo
g 2>&1 & sudo phc2sys -w -m -s ens1f1 -R 8 -f configs/default.cfg > phc2sys.log
2>&1 &
```

Note:

`offset < 100` represent sync

```
## Find task, then you can kill them ps aux | grep ptp4l ps aux | grep phc2sys
```

6.4 Run OAI gNB

```
cd ~/openairinterface5g/cmake_targets/ran_build/build/ sudo ./nr-softmodem -O
../../../targets/PROJECTS/GENERIC-NR-5GC/CONF/oran.fh.band78.fr1.273PRB.conf --s
a --reorder-thread-disable > ~/run_OAI_gNB.log 2>&1
```

- OAI RUN Full logs

7. Appendix

Troubleshooting Guide

- NOTE_LINK

Create Virtual Function

Check NIC

```
ifconfig
```

```
... ens1f1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 ether 40:a6:b7:9
2:c4:75 txqueuelen 1000 (Ethernet) RX packets 0 bytes 0 (0.0 B) RX errors 0 drop
ped 0 overruns 0 frame 0 TX packets 0 bytes 0 (0.0 B) TX errors 0 dropped 0 over
runs 0 carrier 0 collisions 0 ...
```

```
ethtool -i ens1f1
```

```
driver: i40e version: 4.18.0-425.10.1.rt7.220.el8_7.x firmware-version: 8.30 0x8
000a4db 1.2926.0 expansion-rom-version: bus-info: 0000:3b:00.1 supports-statisti
cs: yes supports-test: yes supports-eeprom-access: yes supports-register-dump: y
es supports-priv-flags: yes
```