

5G NFV RAN Network Slicing Bench: The 5th-Generation Network Function Virtualization Radio Access Network Slicing Benchmarks

Jianda Wang
Electrical and Computer
Engineering Department
The University of Texas at Dallas
Richardson, USA
jxw174930@utdallas.edu

Yang Hu
Electrical and Computer
Engineering Department
The University of Texas at Dallas
Richardson, USA
Yang.Hu4@utdallas.edu

Abstract – The 5th generation network is expected to enable a fully connected society and handle a wide range of services in the mobile communication world. The demands of a fully connected society and wide diversity of services are characterized by tremendous growth in connectivity, stringent constraint in reliability and ultra-low latency in transmit activity. Corresponding to the new requirements of applications and services, the 5G system aims to provide a scalable implementation of network services and a flexible platform enabling vertical-structural business model. On this basis, a network slicing enabled Network Function Virtualization (NFV) paradigm emerges as a promising future-proof framework. Network Function Virtualization (NFV) enables scalable implementation of network services on cloud infrastructure. Besides, Network Slicing is an infrastructure adhering to the commercial and technical requirements of different industries. Attentions are heavily paid on core network function virtualization recently, Radio Access Network (RAN) network slicing enabled NFV is seldom highlighted by research field. This paper introduces 5G NFV RAN Network Slicing Bench, a set of workloads that represent common performances of Network Slicing enabled NFV system at Radio Access Network. 5G NFV RAN Network Slicing Bench selects mobile network platforms OpenAirInterface and FlexRAN as its benchmark systems, since OAI and FlexRAN are the most comprehensive open-sourced systems including network slicing enabled eNodeB and EPC (Evolved Packet Core). In the 5G NFV RAN Network Slicing Bench, we also show up the comprehensive workloads characterizations in OAI and FlexRAN systems.

Keywords—5G, NFV, RAN, Network Slicing, Benchmark

I. INTRODUCTION

The dramatic growth of mobile data traffic driven by Internet and smart devices has triggered the 5G for the next generation of mobile telecommunication. The proliferation of connected objects and devices paves the way to a wide range of new services in various industry sectors and vertical markets (e.g. Energy, e-health, smart city, connected cars, industrial manufacturing, etc.) 5G networks aim to provide a scalable and flexible implementation of network services enabling new business cases which integrating vertical industries.

In responding to the next generation network requirements and challenges, 5G Networks explore Network Function Virtualization and network slicing to enhance its functional and architectural viability. Network Function Virtualization (NFV) is a novel paradigm that enables scalable and flexible implementation of network services on cloud infrastructure.

The network slicing feature will satisfy the demand of vertical sectors which request dedicated telecommunication services by providing customer on-demand network slice.

Today, most of work highlights the Network Function Virtualization on core network [1] [2] [3], research concentrating on radio access network is not adequate. The radio access network (RAN) – the expensive and complex part of the mobile network infrastructure, offers great opportunities to benefit from NFV ideas.

Radio Access Network experimentations over testbeds with commercial equipment restricts configuration capabilities and flexible deployment due to constraints imposed by operators and large vendors. This has resulted in the need for an open and flexible radio access network experimentation platform with high degree realism. In recent years, several radio access network emulation projects are proposed [4] [5]. Among open-source solutions, Eurecom's OpenAirinterface (OAI) appears to be the most promising and complete project. OAI allows one to carry out experiments flexibly and provides the possibility to analysis and develop the features of Radio Access Network system.

Network slicing attracts the attention of the research and industrial community recently, network slicing at radio access network side are still in its infancy, even though many architectures and prototypes have been proposed for core network (CN) slicing [6] [7] [8]. FlexRAN, to the best of our knowledge, is the only open-source RAN platform tailored to support network slicing feature. FlexRAN is designed with flexibility, programmability and ease of deployment. FlexRAN offers a degree of flexibility to dynamically realize mobile operations inside base stations.

This paper proposes 5G NFV RAN Network Slicing Bench. 5G NFV RAN Network Slicing Bench selects mobile network platforms OpenAirInterface and FlexRAN as its benchmark systems, since OAI and FlexRAN are the most comprehensive open-sourced systems including network slicing enabled eNodeB and EPC (Evolved Packet Core). The contributions of this paper are as follows: 1) We provide comprehensive workload characterizations that represent the performance of OpenAirInterface and FlexRAN testbeds 2) Through the performance analysis, we put out bottleneck in OpenAirInterface and FlexRAN testbeds which need to be improved in the future work.

II. RELATED WORK

Network Virtualization Networks (NFV) has received substantial attention from the research community in recent years with both academia and industry recognizing its benefits on operational mobile networks. Although most of the work highlights the NFV process on core networks [1] [1] [1] [1], there are still several NFV projects [1] [1] [1] proposed at side the of radio access networks. As a complementary part, the network slicing [1] [1] [1] are always put out as a branch feature under radio access networks virtualization projects.

While the scope of the above mentioned works includes the RAN virtualization and network slicing realization, none of them provide a systematic benchmark for the performance that represent common protocols and functions in RAN networks. In the last few years, several works [1] [1] [1] [1] provide the performance analysis and study on the overall OpenAirInterface system. [1] [2] introduce the concepts and architecture of the OpenAirInterface system. [1] performs a thorough profiling of OAI, in terms of execution time, on the user plane data flow. [2] validates two MAC schedulers and analyzes the OpenAirInterface system, in terms of memory occupancy and execution time

A key limitation of the aforementioned works is that none of them provide comprehensive workloads characterization for OpenAirInterface and FlexRAN system, which serves as the motivation of our work in this paper.

III. 5G NFV RAN NETWORK SLICING BENCH OVERVIEW

This section gives a high-level overview of the OpenAirInterface and FlexRAN platform, the key testbed of 5G NFV RAN Network Slicing Bench. Fig. 1 illustrated 5G NFV RAN Network Slicing Bench architecture.

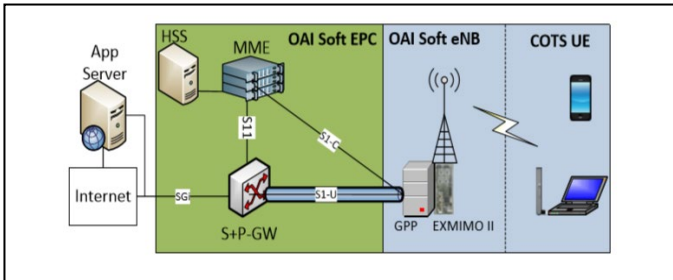


Fig. 1. OAI and FlexRAN system overview

The OpenAirInterface (OAI) is the most complete open-source RAN software experimentation and prototyping platform created by the Mobile Communications Department at EURECOM. The OAI platform includes a full software implementation of mobile cellular systems compliant with 3GPP standards in C under realtime Linux optimized for x86. For the 3GPP Access-Stratum, OAI provides standard-compliant implementations of PHY, MAC, RLC, PDCP and RRC, spanning the entire protocol stack from the physical to networking layer, for both eNB and UE. Moving to the core network, the OAI also comprises of standard compliant implementations of a subset of 3GPP EPC component with the Serving Gateway (S-GW), the Packet Data Network Gateway (P-GW), the Mobility Management Entity (MME), the Home

Subscriber Server (HSS) and the Non-Access Stratum (NAS) protocols. Fig. 2 shows a schematic of implemented software stack in the OAI.

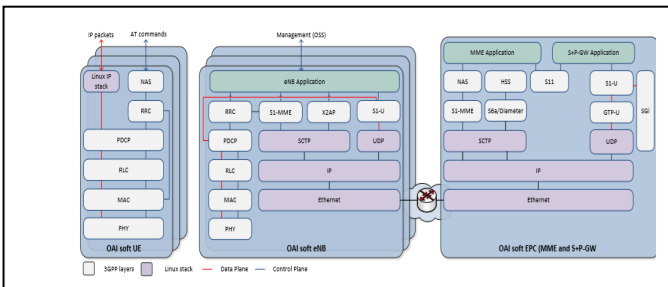


Fig. 2. OpenAirInterface Architecture

FlexRAN is a RAN slicing system design that is in line with the demand of network slicing and the needs of flexible configuration in mobile network. Fig. 3 provides a high-level schematic of the FlexRAN platform, which is made up of two main components: the FlexRAN Service/Control Plane and FlexRAN Application plane. The FlexRAN service and control plane follows a hierarchical design and is composed of a Real-time Controller (RTC) which is connected to a number of underlying RAN runtime, one for each RAN module. The FlexRAN protocol facilitates the communication between the real-time controller and the RAN agent embedded in runtime environment. RAN control applications are realized on the top of the RAN runtime which allowing to coordinate, monitor, and control the state of RAN infrastructure

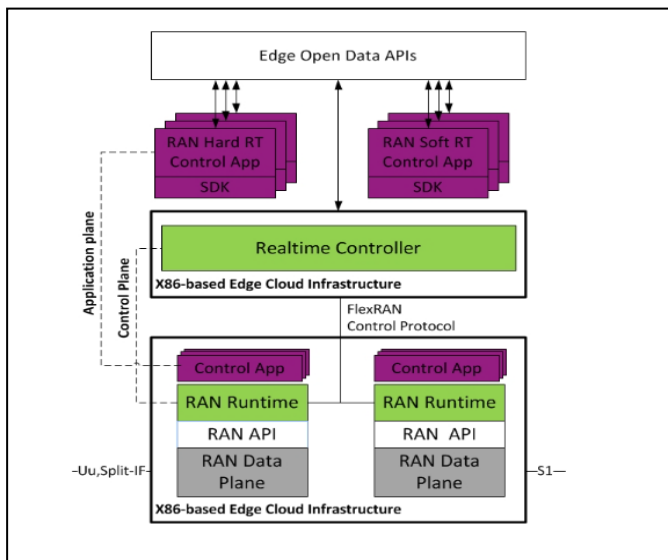


Fig. 3. FlexRAN Architecture

IV. EXPERIMENT SETUP

As illustrated in Figure 4, the experimental testbed consists of one/two units of Commercial Off-The-Shelf (COTS) UE, one unit of OAI eNB and one unit of EPC. We use Intel Core machines (Core i7-8700 @ 3.20GHz 16GB RAM) for eNB, Intel Xeon machine (E5405 @ 2.00GHz 4G RAM) for EPC and Huawei Honor 8 as our UE. FlexRAN is implemented in the same machine with eNB. The eNB version we use is master

branch v0.6.1. For EPC and FlexRAN, we use develop branch. The Operation system used for both machines is Ubuntu 16.04. The testbed is implemented with a real RF front-end (Ettus B210 USRP). All the experiments were conducted with the same eNodeB configuration, namely FDD with 5 MHz bandwidth in band 7. We utilize app ‘Speedtest’ to generate traffic in the UE side.



Fig. 4. OpenAirInterface Experiment Setup

Intel VTune Amplifier is an application used to analyze software performance for x86 based machines using hardware performance counters. For the 5G NFV Network slicing bench, we use Intel VTune Amplifier to profile the benchmarks and collect micro-architectural data.

TABLE I is the source file names we monitored in the experiment related to the LTE protocol Modules and network stack defined by the OAI software.

TABLE I. SOURCE FILES INSIDE OAI MODULE

LTE Protocol Modules and Network Stack	Source File Names
Module Initial	lte_init.c, pss.c, sss.c
Module RRC	L2_interface.c, rrc_common.c, rrc_eNB.c, rrc_eNB_UE_context.c
Module PDCP	pdcp.c, pdcp_fifo.c, pdcp_primitives.c, pdcp_util.c
Module RLC	rlc.c, rlc_am.c, rlc_am_timer_reordering.c, rlc_mac.c, rlc_um.c, rlc_am_timer_poll_retransmit.c, rlc_um_dar.c, rlc_um_reassembly.c, rlc_um_receiver.c, rlc_um_segment.c

Module MAC	eNB_scheduler.c, eNB_scheduler_bch.c, eNB_scheduler_dlsch.c, eNB_scheduler_primitives.c, eNB_scheduler_ulsch.c, preprocessor.c
Module DCI	dci.c, dci_tools.c
Module CRC	crc_byte.c
Module Coding	3gpplte_sse.c, 3gpplte_turbo_decoder_avx2_16bit.c, 3gpplte_turbo_decoder_sse_16bit.c, dlsch_coding.c, ulsch_decoding.c
ModuleRate Matching	lte_rate_matching.c
Module Scrambling	dlsch_scrambling.c, lte_ul_channel_estimation.c
Module Modulation	dlsch_modulation.c, lte_mcs.c, ulsch_demodulation.c
Module OFDM Layer Mapping Precoding Resource Mapping	lte_dfts.c, ofdm_mod.c
Module Control Channel	pbch.c, pcfich.c, phich.c pmch.c prach.c
Module USRP RF	usrp_lib.cpp
Module PHY Procedures Top-level procedures	lte_enb.c, phy_procedures_lte_common.c, phy_procedures_lte_eNb.c
Module User Interface	intertask_interface.c, intertask_interface_dump.c

V. BENCHMARK CHARACTERIZATION

In this section, we provide a detailed description for the performance of OAI and FlexRAN platform.

A. OpenAirInterface and FlexRAN Overall performance

Fig. 5 represents inefficiencies in CPU usage for the overall OpenAirInterface system. We can see that the retiring part is just 43.68%, which is not satisfying. Back-End Bound, Front-End Bound is 37.4% and 23.80%, respectively. Back-End Bound is deteriorated so severely that optimization for the Back-End implementation becomes inevitable. Inside Back-End Bound, Memory Bound is 23.8% and Core Bound is 13.6%. The high memory Bound is mainly caused by L1 Bound and L3 Bound and the high core Bound is mainly caused by Port Utilization. For the Front-End Bound, the Front-End latency is 12.5%, which means that the improvement is needed for OpenAirInterface system to reduce the Front-End latency.

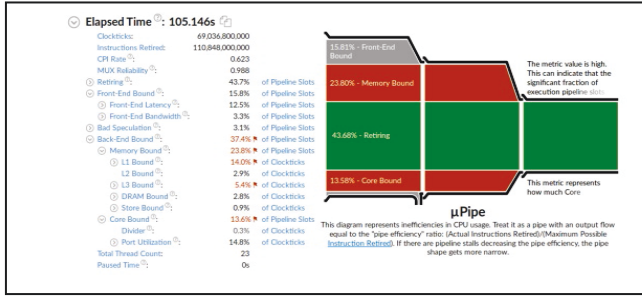


Fig. 5. Example of a figure caption. (figure caption)

B. CPU Time for OpenAirInterface Protocol Modulese

Fig. 6 shows the CPU time spent by each OpenAirInterface module as a fraction of the overall system runtime (the module source files are shown in Table I). We see that the Module OFDM/Layer Mapping/Precoding/Resource Element Mapping takes most of the execution time (more than 45%). The second and third CPU time consuming module is Module DCI and Module Coding/Decoding. Compared to the WiBench, which only simulate the PHY layer for Radio Access Network, the most CPU time consuming part is Turbo decoder. Our result shows that the most CPU time consuming part is the module related to OFDM. The reason for this is that WiBench analysis the UE side, which utilize SC-FDMA algorithm. For OpenAirInterface, the analysis is the eNB side, which employ OFDM algorithm. The ofdm_mod.c(slot generation function) source file is high CPU consuming compared to the module used in WiBench. Thus, for high throughput applications, either Module OFDM/Layer Mapping/Precoding/Resource Element Mapping should be highly optimized for the OpenAirInterface platform.

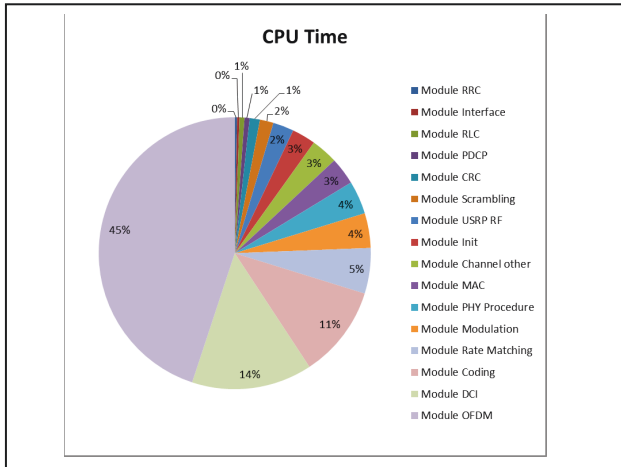


Fig. 6. Example of a figure caption. (figure caption)

C. Instructions Execution

Instruction per cycle (IPC) is a fundamental performance metric indicating the average number of instructions executed for each clock cycle, which is used to measure instruction level parallelism. Fig. 7 shows IPC of each OpenAirInterface Module. We can see that most of Layer 1 Protocols (Physical Layer) have middle-level IPC values, greater than that of Layer 2 Protocols (RRC, PDCP, RLC, MAC). The IPC of the Layer 2 protocol workloads ranges from 0.32 to 1.34. PDCP protocol

has the lowest IPC value among Layer 2 protocols workloads. The IPCs of the Layer 1 protocols have higher IPC values compared to Layer 1 protocols. For example, scrambling has a IPC rate with 3.25 and OFDM has a IPC rate with 3.31. The reason for this is that most of the Layer 1 protocols are computation-intensive, and hence have a higher IPC. While Layer 2 protocols are designed to allocate resources and elements for channels, thus they access memory more frequently compared to Layer 1 protocols. This leads to poor temporal locality, causing long-latency memory accesses for Layer 2 protocols, and hence it has lower IPCs.

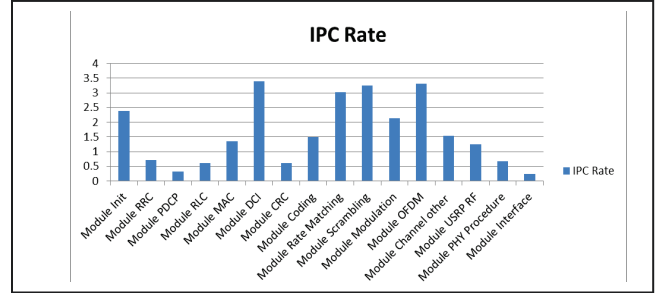


Fig. 7. IPC Rate for OAI Modules

D. Front-End Behaviore

The instruction-fetch stall will prevent core from making forward progress due to lack of instructions. Instruction cache (ICache) and Instruction Translation Look-aside Buffer (ITLB) are two fundamental components, which should be accessed when fetching instructions from memory.

Figure 8 and Figure 9 present the Instruction cache misses (Icache misses) and the Instruction TLB overhead (ITLB overhead), respectively. Layer 2 protocol modules own higher Instruction cache misses than Layer 1 Physical protocol modules (Coding, Rate Matching, Scrambling, Modulation and OFDM procedure). RRC and MAC have large instruction footprints and suffer from severe L1 Instruction cache misses. Higher L1 instruction cache misses result in higher instruction fetch stalls, indicating less efficiency of the front-end. For most of the others benchmarks, the Layer 1 protocols modules instruction cache misses are small, especially the OFDM Procedure Module, whose instruction footprint is relatively rare.

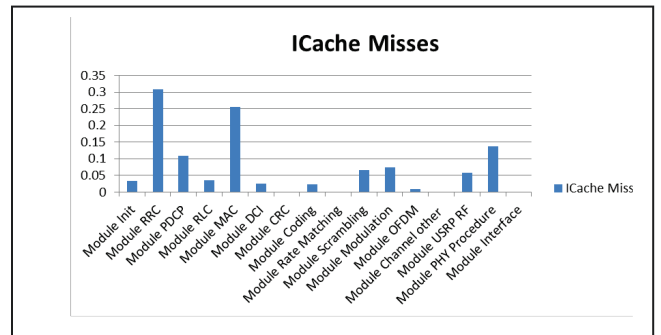


Fig. 8. Icache Misses for OAI modules

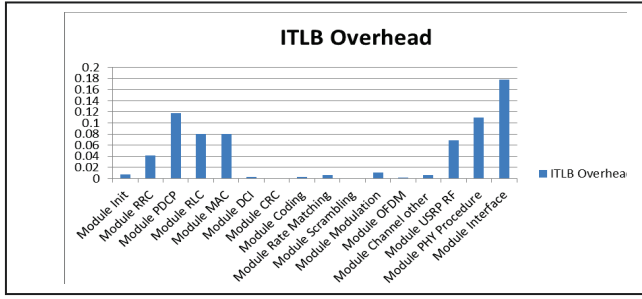


Fig. 9. ITLB Overhead for OAI modules

Consistent with the performance trend of instruction cache misses, Layer 2 protocols modules complete page walks caused by instruction TLB miss are more frequently than that of Coding, Rate Matching, Scrambling, Modulation and OFDM Procedures (Layer 1 Protocols). Page walks will cause a long latency instruction fetch stall, waiting for correct physical addresses so as to fetch instructions, and hence result in inefficiency of front end.

Improving the L1 instruction cache and instruction TLB hit ratios can improve the performance of Layer 1 protocols modules, especially the RRC and MAC module. The third party libraries used by OpenAirInterface system may be the main reason which leads the inefficiency of instruction cache and TLB. Improvement is necessary to enhance the performance of OpenAirInterface in Icache and ITLB aspects.

Figure 10 to Figure 12 lists the information for DSB switches, Front-End Bandwidth MITE and Front-End Bandwidth DSB, which are the main reasons cause the Front-End Bound besides Icache misses and ITLB Overhead.

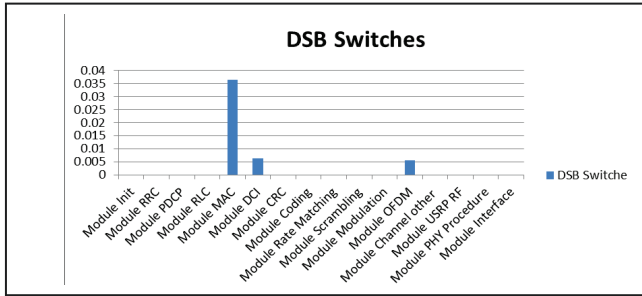


Fig. 10. DSB Switches for OAI modules

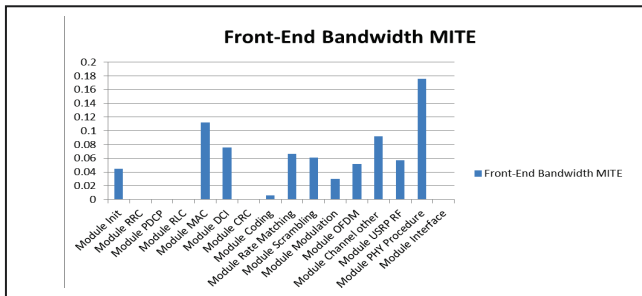


Fig. 11. Front-End Bandwidth MITE for OAI modules

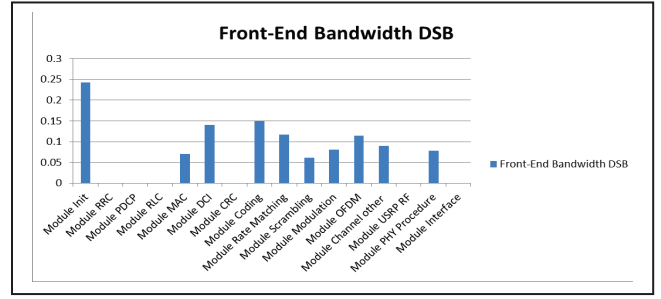


Fig. 12. Front-End Bandwidth DSB for OAI modules

DSB Switches measure the penalty when control flows out of the region cached in the DSB, the front-end incurs a penalty as uOp issue switches from the DSB to the MITE. From the Figure 10, we can see that the DSB switches value for OAI modules is rare, indicating that the OpenAirInterface system can utilize DSB (Decoded Stream Buffer) quite well. Front-End Bandwidth MITE represents a fraction of cycles during which CPU was stalled due to the MITE fetch pipeline issues, such as inefficiencies in the instruction decoders. From Figure 11, we can see that the Front-End Bandwidth MITE value are always below 0.1, except for the module MAC and PHY Procedure. Some optimization may be needed for these two modules. Front-End Bandwidth DSB represents a fraction of cycles during which CPU was likely limited due to DSB (decoded uop cache) fetch pipeline. From Figure 12, we can see that most of the OAI module performance well except the module Init.

E. Back-end Memory Bound - Data Cache and Data TLB Behaviors

The modern processors introduce a deep memory hierarchy to reduce the performance impacts of memory wall. A miss penalty of last-level cache can reach up to several hundred cycles in modern processor. Figure 13 and 14 shows the L1 Bound and L2 Bound for OAI modules, we can see that for most of Layer 2 protocols workloads have lower Data cache misses than that of the Layer 1 protocols workloads. The L1 and L2 cache statistic indicates Layer1 protocols workloads own better locality than the Layer 2 protocols workloads.

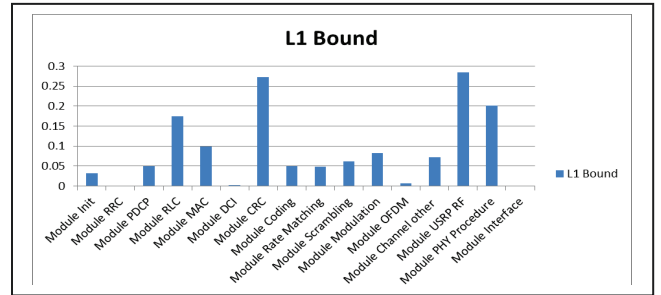


Fig. 13. L1 Bound for OAI modules

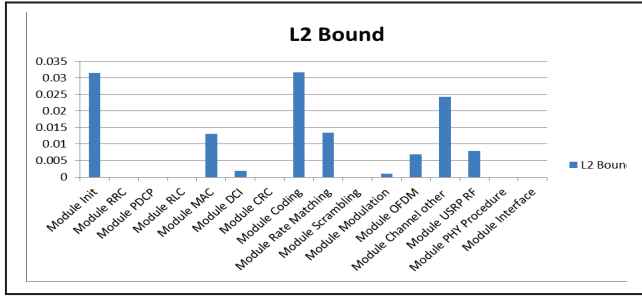


Fig. 14. L2 Bound for OAI modules

From Figure 15, we can find that for both the Layer 1 Protocols workloads and Layer 2 Protocols workloads have low L3 Bound value. We can conclude that for most of Layer 1 Protocols and Layer 2 Protocols, modern processor's LLC is large enough to cache most of data missed from L1 and L2 cache. Figure 16 shows the completed page walks caused by Data TLB misses. For most of the OpenAirInterface workloads with the exception of PHY Procedure and USRP RF Module, the DTLB value is really very rare. The high value for PHY Procedure and USRP RF Module may be the reason which causes the OpenAirInterface System unstable occasionally.

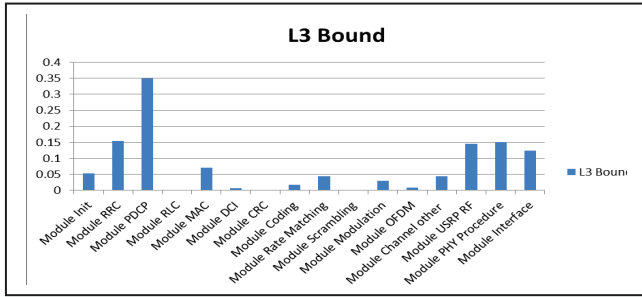


Fig. 15. L3 Bound for OAI modules

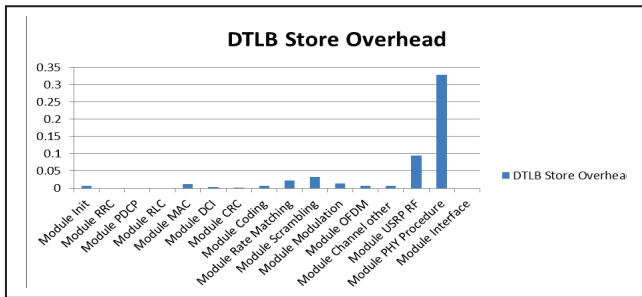


Fig. 16. DTLB Store Overhead for OAI modules

F. Back-end Core Bound – Port Utilization

Backend Bound is mainly caused by the lack of hardware resources (e.g. divider unit) or port utilization because of The instruction dependencies and execution unit overload. In this section, we only present Port Utilization for OpenAirInterface system because the divider bound value is nearly 0 for each OAI Module. From Fig. 17, we find that most of the OAI modules suffer from Port Utilization Bound, indicating that the Port Utilization need to be optimized for each OAI module.

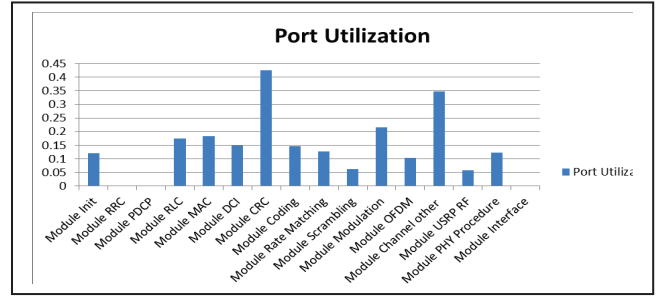


Fig. 17. Port Utilization for OAI modules

VI. THE SUMMARY OF RAN NFV NETWORK SLICING BENCH

Radio Access Network Network Function Virtualization and Network Slicing attract great attention from both academia and industry recently. Since Benchmarks, as the foundation of quantitative design approach, are used to evaluate the systems and new features, we provide the 5G RAN NFV Network Slicing Bench in this paper.

According to our workload characterization work, the RAN modules in the same Layer shared inherent characteristics while owns different properties with other RAN protocol modules. Besides, we show up bottlenecks in several OAI protocol modules, which need to be optimized in the future work.

REFERENCES

- [1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529–551, April 1955. (references)
- [2] J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in *Magnetism*, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
- [4] K. Elissa, "Title of paper if known," unpublished.
- [5] R. Nicole, "Title of paper with only first word capitalized," *J. Name Stand. Abbrev.*, in press.
- [6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Transl. J. Magn. Japan*, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].
- [7] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.
- [8] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529–551, April 1955. (references)
- [9] J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [10] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in *Magnetism*, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
- [11] K. Elissa, "Title of paper if known," unpublished.
- [12] R. Nicole, "Title of paper with only first word capitalized," *J. Name Stand. Abbrev.*, in press.
- [13] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Transl. J. Magn. Japan*, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].

- [14] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.
- [15] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529–551, April 1955. (*references*)
- [16] J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [17] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in *Magnetism*, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
- [18] K. Elissa, "Title of paper if known," unpublished.
- [19] R. Nicole, "Title of paper with only first word capitalized," *J. Name Stand. Abbrev.*, in press.
- [20] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Transl. J. Magn. Japan*, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].
- [21] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.