Resource Profiling for Virtualized Radio Access Networks

Rajashekhar Reddy Tella*, Utkarsh Gupta*, Gudepu Venkateswarlu*, Venkatarami Reddy Chintapalli†
Bheemarjuna Reddy Tamma‡, Koteswararao Kondepu*
* Department of Computer Science and Engineering, IIT Dharwad, Dharwad, India
† Department of Computer Science and Engineering, National Institute of Technology Calicut, Kerala, India
‡ Department of Computer Science and Engineering, IIT Hyderabad, Hyderabad, India
Email: {200030058, k.kondepu}@iitdh.ac.in

***Abstract*—The mobile networks are rapidly evolving, transitioning from specialized hardware to fully virtualized platforms that run on commercial off-the-shelf (COTS) hardware powered by general-purpose processors. Mobile operators are increasingly adopting Network Functions Virtualization (NFV) to leverage the benefits of virtualization, such as ease of deployment, flexibility, and cost savings. A crucial component of this transformation is the implementation of virtualized Radio Access Network (vRAN) solutions, which provide a cost-effective way to deploy the 5G and beyond RAN as a containerized network function (CNF) on COTS hardware. vRAN is more efficient than traditional RAN, as it allows multiple base station workloads to be multiplexed on the same hardware, potentially co-located with other workloads. However, ensuring consistent performance of vRAN workloads in this consolidated environment is challenging due to the high variability and unpredictability arising from contentions for shared system resources such as *CPU cores*, *Last Level Cache (LLC)*, and *Memory Bandwidth*. Thus, there is a strong need to understand the vRAN components if they are affected by any system resource. This paper presents an understanding of how the next-generation scalable processors can be used to run the vRAN components and presents a detailed analysis of the controllable knobs such as *CPU*, *LLC*, *Memory Bandwidth*, and *energy consumption*. Additionally, this work offers an insightful glimpse into the power consumption of various vRAN architectures. The proposed vRAN profiling shows that the observed throughput varies significantly based on the number of pinned CPU cores (e.g., up to $250\ Mbits/sec$), and also similar performance observed with other system resources — LLC allocation and Memory Bandwidth.**

***Index Terms*—Resource Profiling, Virtualized Radio Access Network, Resource consumption, Kubernetes**

I. Introduction

With the advent of Fifth-generation and beyond (B5G) networks for mobile cellular communications, broadband usage increases rapidly due to the growing number of subscribers and the wide range of applications they use [1]. The increase in user service demands necessitates the assurance of Quality-of-Service (QoS), ensuring agile, resilient networks capable of supporting high volumes of data traffic and various dynamic service demands. Mobile network operators aim to meet the emerging dynamic service demands while optimizing capital expenditure (CAPEX) and operational expenditure (OPEX). Virtualization of network functions (VNFs) emerges as an effective solution, offering benefits such as ease of deployment, flexibility, and cost savings [2]. VNFs approach is increasingly applied to mobile networks, particularly in the Radio Access Network (RAN), which is crucial for transmitting data packets to the wireless radio and managing complex signal processing tasks.

Virtualized Radio Access Network (vRAN) offers several advantages but also poses technical challenges. Some advantages of vRAN include — mitigating vendor lock-in, allowing flexible upgrades, facilitating rapid deployment of new standards and services, and the potential to reduce costs [1]. However, it introduces a different energy consumption profiling when compared to traditional Base Stations (BSs) that rely on dedicated hardware. The energy consumption of virtual Base Stations (vBSs) is influenced by factors such as network state (e.g., traffic load and Signal-to-Noise Ratio, or SNR), General Purpose Processors (GPP), and the software implementation of the radio stack. When the vRAN network functions run at the edge, the configuration of edge services (e.g., QoS) and the network (e.g., channel capacity) are closely interconnected, impacting the system's overall service performance and power consumption. Given that RAN accounts for approximately 75% of a mobile network's total energy consumption, addressing these challenges through integrated evaluation and orchestration yields significant improvements in both the performance and energy efficiency of the vRAN [3]. The efficient RAN operations contribute not only to reducing the carbon footprint of ICT networks but also offer economic benefits by optimizing resource as well as energy usage.

[4] presents a system that dynamically adjusts both system and radio resources to ensure Key Performance Indicators (KPIs) for virtualized Radio Access Networks (vRANs). In [5], the power consumption was monitored during the user registration and authentication of various deployed open source 5G Core Networks (5G CN). In [6], a hardware tool is used (i.e., Meross MSS310) to measure the power consumption at the 5G CN. However, these works focus only on power measurement at the 5G CN. In addition, [7] presents an optimization method for power consumption of the RAN, Whereas, [8] provides the integration of software tools with the RAN scenarios to measure the energy consumption. In addition, [9] integrates a software tool (i.e., KEPLER) to measure energy over the Kubernetes (K8s) based RAN scenarios.

The above-mentioned works have not focused on measuring different system resources and their corresponding impact on various system parameters. Note that considering system parameters — Power consumption, CPU consumption, Memory usage, and Cache allocation — are also important for the vRAN while serving various user traffic demands. Measuring and monitoring the key parameters (i.e., resource profiling) of the vRAN plays an important role in optimizing resource utilization without compromising the overall network performance.

This paper focuses on experimentally measuring and monitoring the key system parameters that can enhance the RAN performance by integrating the open-source software tool — KEPLER [10] and understanding the importance of next-generation scalable processors.

The key contributions of the paper are as follows:

- Integration of the open-source software tool with various scenarios of the vRAN to measure system resource profile.
- Experimental evaluation of all the vRAN scenarios with OpenAirInterface (OAI) [11] 5G setup.
- Analysis of the vRAN resource profiling on scalable processors as a function of observed throughput with different performance metrics impact — CPU cores, Cache allocation, Memory bandwidth utilization.

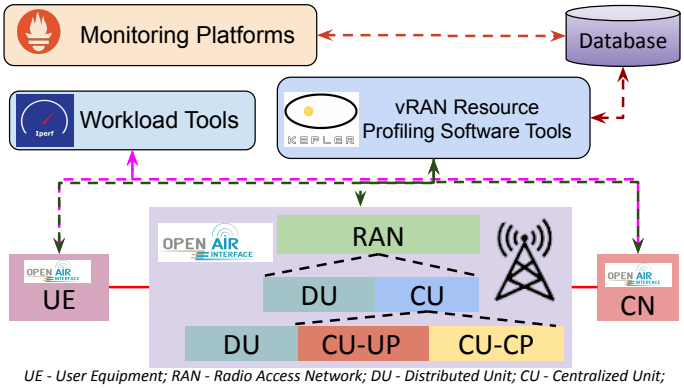

Figure 1: System Architecture

II. SYSTEM ARCHITECTURE

This section outlines the system model, emphasizing the main components, including the Kubernetes (K8s)-based 5G system architecture, vRAN resource profiling software tools, and monitoring platforms. The 5G system architecture comprises three essential elements: User Equipment (UE), Radio Access Network (RAN), and the 5G Core Network (5G CN), as illustrated in Fig. 1. These components collaborate to deliver network services and functionalities.

The Radio Access Network (RAN) segment has evolved significantly, resulting in three primary deployment models to meet the diverse demands of 5G networks:

- *Monolithic RAN:* In this setup, all the RAN functionalities are consolidated into a single entity, known as the Next-generation NodeB (gNB). The gNB includes critical functions like the Physical layer, Medium Access Control (MAC) layer, and Radio Link Control (RLC) layer.
- *Dis-aggregated RAN:* This RAN architecture divides the gNB into distributed units (DUs) and centralized units (CUs). The dis-aggregation of gNB allows for greater flexibility and scalability, enabling operators to optimize network performance by deploying DUs closer to the user edge while centralizing the CUs.
- *Control and User Plane Separation RAN (CUPS RAN):* The CUPS RAN architecture takes dis-aggregation further by splitting the centralized unit (CU) into control plane (CP) and user plane (UP) components. The CP handles signaling and control tasks, such as managing connection procedures, while the UP is responsible for forwarding user data packets. The separation in the CU enhances scalability, allows for independent scaling and customization of control and user planes, and improves operational efficiency by enabling geographical distribution of CP and UP components.

As the RAN deployment architectures evolve, the need for efficient resource management becomes critical. Virtualized RAN (vRAN) systems, running on commercial off-the-shelf (COTS) hardware, require careful profiling of resources such as CPU cores, cache allocation, memory bandwidth, and energy consumption. vRAN resource profiling software tools are essential for assessing these metrics within the 5G system architecture to provide valuable insights into infrastructure resource usage and the associated carbon footprint.

Data collected on vRAN resource utilization is stored in a repository and analyzed using various monitoring platforms. These tools, deployed on the Kubernetes (K8s) master node, integrate seamlessly with the 5G system, enabling operators to optimize resource allocation and reduce operational costs while maintaining high performance and efficiency across different RAN deployment models.

In this study, the power measurement tool Kubernetes Efficient Power Level Exporter (Kepler) is employed. Kepler utilizes software counters and hardware sources such as Running Average Power Limit (RAPL), Advanced Configuration and Power Interface (ACPI), and NVIDIA Graphics Processing Unit (GPU) to monitor power consumption within the K8s master node. Overall, the data from Kepler provides a comprehensive view of power consumption dynamics across the various components of the 5G system.

Table I: 5G Network Parameters

Description	Value
NR Release - Band - Freq.	3GPP Release 16 - Band 78 - 3.6 GHz
RAN type	5G standalone gNB
CU/DU split	Option 2
Physical Resource Block	106
Radio Channel Bandwidth	40 MHz
Midhaul/Backhaul Capacity	1 Gbps Ethernet
UE	OAI based 5G SA UE

III. EXPERIMENTAL SETUP AND RESULTS

A K8s cluster with one node (one master node) has been deployed in R750 Xenon scalable processors servers. This single-node cluster serves as the testing ground for evaluating all three RAN deployment schemes: Monolithic, Disaggregated,

Table II: CPU configuration

Component	Description
Processor	Intel(R) R750 Xeon(R) Platinum 8362 CPU @ 2.80GHz 64 Core Dual NUMA Socket
OS and Kernel	Ubuntu 22.04.5 LTS; 6.5.0-45-generic
Cache	5 MiB L1; 80 MiB L2; 96 MiB LLC
LLC-ways	12
Main Memory	503 Gi

and CUPS. Table II and Table I details the hardware configuration of the K8s cluster and the 5G network parameters, respectively. The 5G components across the three RAN scenarios are deployed using OpenAirInterface [11]. Power consumption measurements across all RAN scenarios are conducted using the Kepler tool, which has been deployed in the K8s cluster.

Power metrics are visualized through Prometheus [12] and Grafana [13]. Prometheus, a widely adopted open-source monitoring tool, collects system resource metrics from various exporters, while Grafana complements Prometheus by providing a user-friendly platform for creating interactive dashboards.

This paper mainly focuses on profiling the resource utilization and power consumption of the RAN (gNB, DU) network function deployed as k8s pods under various workloads. We have used the *iperf3* [14] tool to generate the workload between the UE and the Core Network and to observe power consumption across all deployment schemes and during idle states before workload initiation. In all experiments, ten independent runs are performed and derived confidence intervals at a level of 90%.

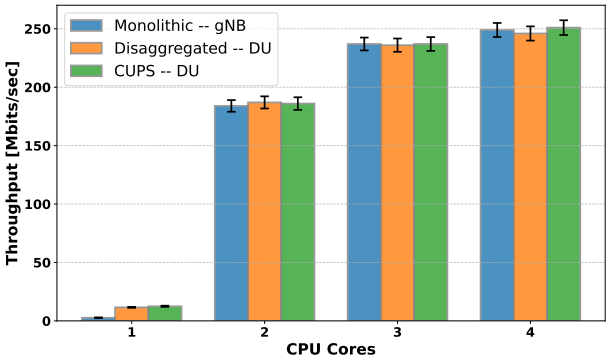

Figure 2: Throughput vs Number of CPU cores

A. CPU cores

The Master node has 128 CPU cores distributed across 2 NUMA nodes. The performance of any function depends on the hardware resources it can utilize. To determine the optimal CPU quantity for a given throughput, we exclusively allocated CPUs to the RAN pod in the cluster. In addition, CPU binding prevents context switching in the running process when a CPU throttle occurs and thus does not affect the performance of the process while running besides any other competing tasks. In this work, we bind the CPU to different numbers of cores ranging from $1-4$ and observe the maximum throughput that can be achieved between the UE and the Core Network.

Fig. 2 shows the throughput as a function number of CPU cores pinned during the experimentation. Initially, CPU core is set to 1 and then later increased up to 4 cores to observe the performance improvement in throughput from UE to Core Network via the RAN pod. We can observe that throughput increases with the number of cores increase, however, after 3 CPU cores, the throughput is not very much. This shows us that by allocating more than 3 CPU cores to the RAN components, does not provide any significant difference. Here, the cache ways are set to 12 for all 128 CPU cores, and no exclusive allotment has been provided. The maximum throughput achieved in monolithic is 249 Mb/sec, dis-aggregated case is 246 Mb/sec, and in the scenario of CUPS is 251 Mb/sec.

Table III shows offered CPU cores and actual consumed CPU millicores as a function of different deployment scenarios. We can observe that in all the considered scenarios around 3000 millicores are used even when we set the 4000 millicores during the experimentation. This shows allocating more number cores may not be much useful after some inflection point, and this is confirmed as shown in Fig. 2.

Table III: Offered CPUs Cores vs Actual Consumed Cores

Scenario	Offered CPUs millicore [m]	Actual Consumed CPU millicore [m]
Monolithic	1000	968
Disaggregated		955
CUPS		954
Monolithic	2000	1955
Disaggregated		1931
CUPS		1887
Monolithic	3000	2644
Disaggregated		2596
CUPS		2419
Monolithic	4000	3033
Disaggregated		3026
CUPS		3017

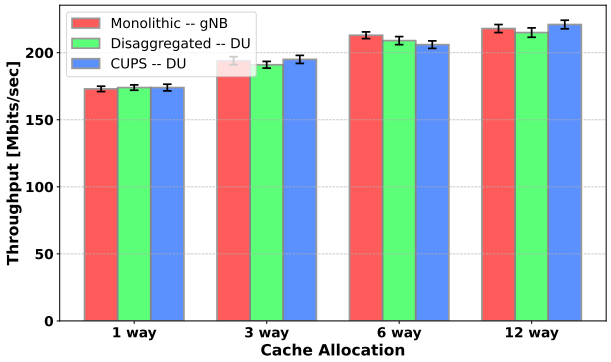

Figure 3: Throughput vs Cache Allocation

B. Last Level Cache ways

Last level cache (LLC) refers to the highest-level cache in a computing system, usually shared by all functional units on the chip, such as CPU cores, integrated graphics processors (IGP), and digital signal processors (DSP). The LLC is a buffer between the high-speed processor cores and the relatively slow main memory. If the cache size is high, then recently used data can be put in that and later used when needed instead of going to main memory, thereby saving time and improving the performance.

In the experimental setup, the nodes have a 12-way cache, each 8 MB summing up to 96 MB LLC. We have allocated 4 CPU cores in all 3 scenarios, categorized the 4 CPUs into

a Class Of Service (COS), and allocated the cache ways to it. We have varied the allocated cache ways from 1,3,6 and 12 ways and observed the maximum achievable throughput between the UE and Core Network. Since the size of a single cache way is also significantly high i.e, 8 MB, we can see that the performance of 1 way cache with 4 CPUs is comparable with 12 way cache with 4 CPUs. So, if there is a high demand for cache, one can use a higher number of CPUs at a lesser cache way to achieve the same performance. As expected, the throughput increases with the increase in cache ways allocated to the RAN pods and can be observed in Fig. 3.

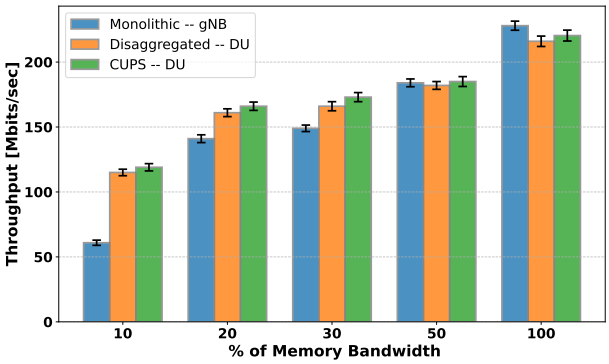

Figure 4: Throughput vs Memory Bandwidth

C. Memory Bandwidth

Memory bandwidth refers to the rate at which data can be transferred to/from a computer's memory (RAM) per unit of time. Memory Bandwidth is a crucial factor in determining the overall performance of a computer system, as it directly impacts the speed at which a CPU can access and process data. Increasing memory bandwidth often requires higher power consumption. To determine the optimal memory bandwidth for a given throughput, we varied the memory bandwidth and measured the maximum achievable throughput. The CPU cores are set to 4 (pinned), and the cache ways are set to 12. In the case of dis-aggregated and CUPS (Fig. 5, the throughput at 10% memory bandwidth is half the throughput at the 100% memory bandwidth, whereas in monolithic, it is fourth.

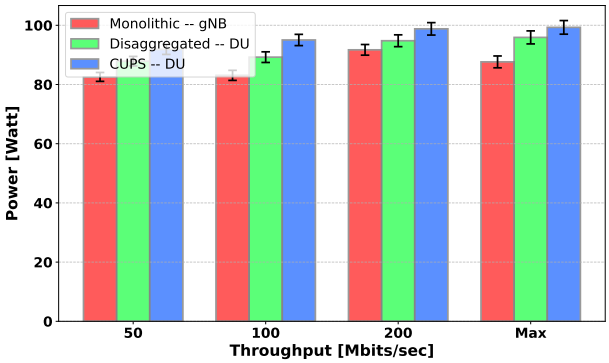

Figure 5: Power vs Throughput

D. Power

The Kepler tool is used to measure the power consumed by the RAN pod under various workloads generated using *iperf3*. Similar to the above experiment, The CPU cores are set to 4 (pinned), and the cache ways are set to 12. We measured power consumption by varying throughput workloads while isolating the RAN pod through dedicated 4 CPU cores allocation in all three scenarios of RAN. We monitored the power consumed by this RAN pod over a 10-minute interval for each throughput load. We have observed that the CUPs scenario consumes more power compared to the remaining two for a given workload.

IV. Conclusion and Future Works

In this paper we have profiled the RAN component in the cellular networks by analyzing CPU utilization, LLC (Last Level Cache) ways allocation, and memory bandwidth to different workloads. The results shown that the considered performance metrics as strong influence on achieving throughput and the optical resources can be allocated to achieve better performance among the considered parameters. In addition, open-source tools are utilized (Kepler) to measure the power consumed by RAN under various workloads and scenarios. Future work would focus more on the power aspect of RAN, and come up with power-efficient configurations.

References

[1] RedHat, "What is vRAN?" https://www.redhat.com/en/topics/virtualization/what-is-vran, 2018, Accessed: August 10, 2024.

[2] V. R. Chintapalli, S. B. Korrapati, M. Adeppady, B. R. Tamma, B. R. Killi *et al.*, "NFVPermit: Toward Ensuring Performance Isolation in NFV-Based Systems," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1717–1732, 2023.

[3] L. Larsen, H. Christiansen, S. Ruepp, and M. Berger, "Towards Greener 5G and Beyond Radio Access Networks - A Survey," *IEEE Open Journal of the Communications Society*, vol. PP, pp. 1–1, 01 2023.

[4] K. A. Malde, V. R. Chintapalli, B. Sharma, B. R. Tamma, and A. Antony Franklin, "Jars: A joint allocation of radio and system resources for virtualized radio access networks," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, 2023, pp. 1–9.

[5] G. Lando, L. A. F. Schierholt, M. P. Milesi, and J. A. Wickboldt, "Evaluating the Performance of Open Source Software Implementations of the 5G Network Core," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2023, pp. 1–7.

[6] A. Bellin, M. Centenaro, and F. Granelli, "A Preliminary Study on the Power Consumption of Virtualized Edge 5G Core Networks," in *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*. IEEE, 2023, pp. 420–425.

[7] A. Abrol and R. K. Jha, "Power Optimization in 5G Networks: A Step Towards GrEEn Communication," *IEEE Access*, vol. 4, pp. 1355–1374, 2016.

[8] V. Gudepu, B. Chirumamilla, R. R. Tella, A. Bhattacharyya, S. Agarwal, L. Malakalapalli, C. Centofanti, J. Santos, and K. Kondepu, "EARNEST: Experimental Analysis of RAN Energy with Open-Source Software Tools," in *2024 16th International Conference on COMmunication Systems NETworkS (COMSNETS)*, 2024, pp. 1148–1153.

[9] V. Gudepu, R. R. Tella, C. Centofanti, J. Santos, A. Marotta, and K. Kondepu, "Demonstrating the Energy Consumption of Radio Access Networks in Container Clouds," in *NOMS2024, the IEEE/IFIP Network Operations and Management Symposium*, 2024.

[10] Uptime Engineering, Inc., "Kepler: Kubernetes Energy Meter," https://kepler.dev/, 2021, Accessed: April 22, 2024.

[11] OpenAirInterface Software Alliance, "OpenAirInterface," https://www.openairinterface.org/, 2007, Accessed: April 22, 2024.

[12] Prometheus Authors, "Prometheus: The Prometheus Monitoring System," https://prometheus.io/, 2012, Accessed: April 22, 2024.

[13] Grafana Labs, "Grafana," https://grafana.com/, 2014, Accessed: April 22, 2024.

[14] iperf3, "iperf3 Download Page," n.d., Accessed: Date. [Online]. Available: https://iperf.fr/iperf-download.php