

An ASIP Based Physical Layer Virtualization Method of Centralized Radio Access Network

Fang Xiao^{**}, Yiqing Zhou^{*}, Shan Huang^{†*}, Jiangnan Lin^{†*}, Lin Liu^{*}

Beijing Key Laboratory of Mobile Computing and Pervasive Device,

^{*}Institute of Computing Technology, Chinese Academy of Sciences,

[†]University of Chinese Academy of Sciences,

Beijing, China

xiaofang@ict.ac.cn

Abstract—With the rapid increasing demand on the data traffic and energy efficiency, centralized radio access network architecture is proposed for next generation communication. Similar to the VM (Virtual Machine) running on the cloud computer servers, V-BS (Virtual Base Station) is proposed. Problem exists since characteristics of V-BS PHY layer tasks are different from traditional computing tasks in the cloud-computing, such as time constraint, data dependency and heterogeneous mapping. This paper proposed a new virtualization method based on ASIP (Application Specified Instruction Set Processor) architecture, with consideration of both scheduling efficiency and resource utilizing. Experiments are made to evaluate the validation of the new virtualization method.

Keywords—virtualization; physical layer; centralized radio access network; ASIP

I. INTRODUCTION

Virtualization technology has been established for decades, and with consolidation and load balancing features in [1] and [2]. With increasing demands of the mobile communication network, centralized radio access network has been proposed to improve networking costs and provide increased flexibility, as in [3], [5] and [6]. Similar to VM, Virtual Base Station (V-BS) is proposed, which identifies the base station program running on the dynamically allocated hardware resources, as introduced in [3], [4] and [5]. Baseband Unit Pool (BBU pool) is defined as the hardware resources pool for V-BS. Consolidation and load balancing features are also required to enhance hardware resource utilization and system robustness in BBU pool. However, different characteristics of V-BS PHY layer tasks determine that traditional virtualization technologies are no suitable and applicable.

PHY layer task is a stream signal processing work with multiple stages, data dependencies and time constrains, e.g. LTE PHY layer task. This complexity contributes to the difficulty of running PHY layer tasks on the operation system based Virtual Machine. In previous studies, two architectures are proposed for LTE V-BS PHY layer tasks: (1) ASIP (Application Specific Instruction Set Processor) based architecture, as in [6]; (2) GPP based Software Define Radio (SDR) architecture with OS, as in [7]. As far as limitations existed in real-time responding performance and calculating efficiency of PHY layer tasks with GPP based OS, which is

introduced in [3] and [8], ASIP based architecture is chosen in this paper.

In this paper we firstly introduce the necessity of PHY layer virtualization in the centralized radio access network. Next we analyze the feature of the LTE PHY layer tasks, and divide the DAG into static part and dynamic part. Then we propose a four step method to virtualize the ASIP hardware resources based on the characteristics of LTE PHY layer tasks with a new refine algorithm. At last, simulation experiments are executed to estimate the performance of the proposed algorithms.

II. FEATURE ANALYZE

A. PHY layer virtualization target

While physical layer is running on ASIP array and upper layer protocol is running on GPP array, the network architecture is shown in Fig.1. In this architecture, ASIP based SOC is connect to RRH (Remote Radio Header) through a high speed digital IQ data switch.

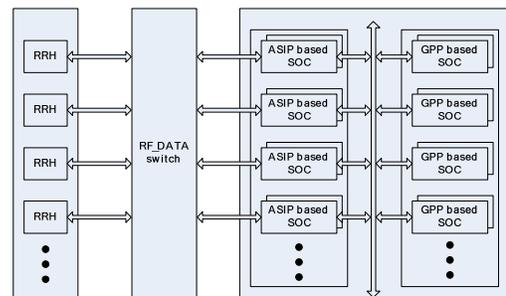


Fig. 1. Architecture of ASIP based centralized radio access network.

LTE PHY layer task can be defined as a direct acyclic graph (DAG) with wireless algorithm kernels in different processing stages, and can be divided into two parts: downlink task and uplink task. Subject to the low efficiency of soft channel coding and de-coding, as in [14], hardware accelerators are adopted in the ASIP based architecture.

Therefore, the PHY layer virtualization problem can be summarized as to schedule multiple DAG graphs above to ASIP arrays dynamically with consideration of hardware utilize efficiency, while satisfies real time constraints. The time constraint relationship is shown in Fig.2.

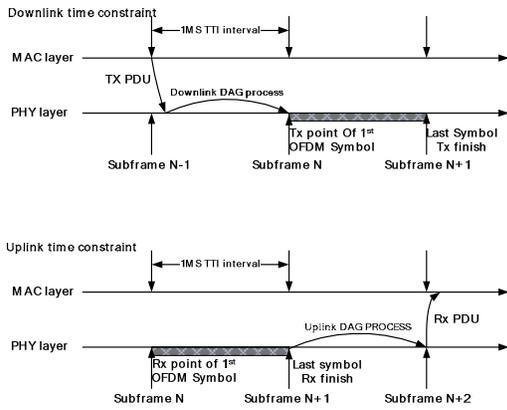


Fig. 2. Time constraint of LTE downlink and uplink.

B. Hardware Structure Analyze

A hierarchical tree structure is adopted for ASIP topology. Leaves represent the ASIP nodes in the hardware resource pool and branches represent the switches. ASIP is the elementary component of the computing pool, and can be characterized by control processors, DSPs and hardware accelerators.

C. DAG Characteristic Analyze

As mentioned in II.A and [9], both the uplink DAG and the downlink DAG can be divided into a static sub-graph and a dynamic sub-graph. In static sub-graph, pre-scheduling can be determined with corresponding amount of hardware resources.

The dynamic sub-graph should be more concerned, for the reason that the make span changes drastically since the UE number and MCS number varies with time, e.g. the execution time variation of PUSCH FEC is shown in Tab. I, which is tested on the ORX (Outer RX) module of DX-005-01 MP-SoC, as mentioned in [15]. Therefore, dynamic scheduling should be realized to utilize the spare resources of the ASIP SOC when the V-BS PHY layer load is low and to explore new spaces on other ASIPs when overload will occur.

TABLE I. FEC DECODING EXECUTION TIMES OF DIFFERENT TBS AND PRB

TBS configure for 1 UE	PRB number	PUSCH FEC de-coding time consumption /ms
32	2	0.007
1384	50	0.229
12960	50	0.442
25456	100	0.896

III. ALGORITHM DESCRIPTION

In previous study, many DAG algorithms are proposed. Pipeline scheduling algorithms intend to divide both DAG and hardware resources into several stages for pipelining execution, as in [10] and [11]. List scheduling algorithms use heuristic methods to meet the scheduling problem which has been proved NP (Non-deterministic Polynomial), as in [12] and [13]. However, most of above methods are not applicable when scheduling multiple LTE DAGs on computing clusters.

Thus, a specific scheduling method for LTE PHY layer virtualization is proposed to maximize the utilization efficiency of the hardware resources, while satisfying real time and bandwidth constraints. This new algorithm is called *Increment Scheduling with Pre-defined Parallelism*. This method is executed in following 4 steps. In order to avoid verbose description, only uplink DAG will be discussed in step 1 and step 2 in this paper, downlink DAG can be analyzed by similar method.

A. Re-model DAG and hardware bonding

DX-001-05 MP-SoC is chosen as the basic ASIP architecture, as mentioned in [15]. In this ASIP, DSP can be used for various LTE algorithm kernels, OTX (Outer TX) and ORX (Outer RX) are specified hardware co-processors for channel FEC processing.

Therefore, in dynamic sub-graph, as mention in II.A, FEC algorithm nodes such as CRC, Turbo and Rate-Matching of the same channel are merged to one node and mapped to the OTX for the downlink and ORX for the uplink. The total execution time of ORX or OTX for the specified channel can be pre-calculated based on V-BS configuration, e.g. the execute time of PUSCH FEC running on the ORX can be simply formulated in (1). F_1 is the factor to reflect how the number of PRB influences the final execution time, and is decided by the performance of the channel De-scrambler, De-interleaver and De-rate Matching components. F_2 indicates how the number of TBS influences the final execution time, and is decided by the performance of the channel Turbo De-coder and CRC components. F_3 is a fixed overhead of each UE.

$$T_{exe} = \sum_{i=1}^{N_{UE}} Torx_i, Torx_i = PRB_i \cdot F_1 + TBS_i \cdot F_2 + F_3 \quad (1)$$

In static sub-graph, nodes are mapped to the DSP processor, with fixed resources requirements. Since only the V-BS bandwidth influences the execution time of the static node, 5M Hz, 10M Hz and 20M Hz bandwidth is discussed.

TABLE II. STATISTICS OF INNER RX WITH DIFFERENT BANDWIDTH

Execution Cycles	5M HZ	10M Hz	20M Hz
Remove CP	149	237	413
Derot	177	287	507
FFT	510	1140	2305
Dexplace	200	272	416
RE-demap (with hopping)	6259	8834	13984
CE(LS)	605	1080	2005
EQU(SIMO-MMSE)	1874	3402	5851
IDFT	5595	9840	9848
DeQAM64	2284	4129	8024
Total cycle number	17653	29221	43353

B. Parallelism explore for most time consuming nodes

1) PUSCH FEC decoding

In PUSCH decoding processing, data from different UE can be executed parallel on different ORX s of the different ASIPs, with no data dependency from the start of PUSCH decoding to the end, as shown in Fig. 3.

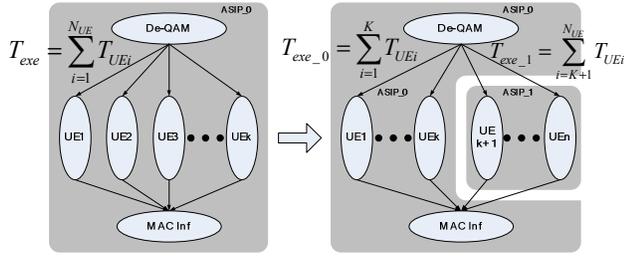


Fig. 3. SPLITTED PUSCH decoding mapped on different ASIPs

2) Static processing for OFDM symbols

Unlike FEC PUSCH decoding nodes, the static symbol processing is parallelized by OFDM symbol, which arrive sequentially in time. A straight parallelized is adopted, as in Fig. 4.

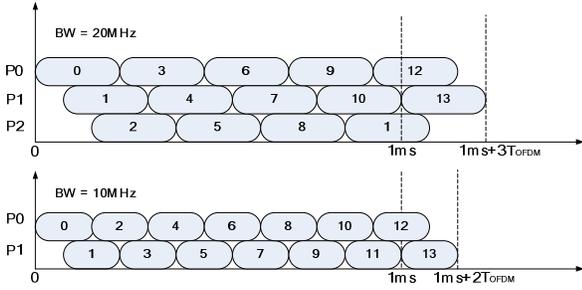


Fig. 4. Parallel scheduling of smybol processing with different bandwidth

C. Initial mapping with redundancy

The initial map pattern will affect the convergence time of the refine algorithm. This is not concerned since the centralized radio access network is running continuously. Therefore, V-BS PHY DAGs are homogenously mapped to the local clusters.

D. Refine algorithm to enhance utilization

With the refine process, PHY layer DAGs on low workload ASIPs will consolidate to another ASIP in order to reduce the number of active ASIPs. When a workload of an ASIP is increased beyond the real-time boundary, some DAG on this ASIP will be immigrated to another ASIP partially or entirely.

This algorithm is based on the following models:

1) N_a ASIP chips in a cluster and N_c local clusters.

2) Q DAGs already mapped to the $N_a \cdot N_c$ ASIPs.

3) The load of each DAG is represent as $R(i) = [\alpha_i, \beta_i, \gamma_i]$. i indicates i_{th} DAG. α, β, γ is the utility ratio of ORX, OTX and DSP cores, as in (3), and can be calculated through LTE characteristics, as described in (1) and Fig. 3.

$$\alpha = \frac{\sum_{n=1}^{N_{UE}} T_{orx_n}}{T_{total}}, \beta = \frac{\sum_{n=1}^{N_{UE}} T_{otx_n}}{T_{total}}, \gamma = \frac{N_{core}}{N_{total}} \quad (3)$$

4) ASIP load is represent as $S(j,k) = [A_{jlo}, B_{jlo}, Y_{jk}]$. $S(j,k)$ indicates the j_{th} ASIP in the k_{th} local cluster. The relationship between $S[j,k]$ and $R[i]$ is shown in (4).

$$S(j,k) = \sum_{DAG_i \in ASIP_{jk}} R(i) \quad (4)$$

5) The guard line S_{guard} is designed to trigger load balance behavior.

The algorithm with designed heuristics is shown in Fig. 5.

Refine Algorithm

- 1) Acquire the initial map list of DAGs and active ASIPs.
- 2) While in each iteration:
- 3) Calc each element R and S .
- 4) If overload happen on one asip, then immigrate tasks to other local asip.
- 5) If no corrspnd asip for whole task, immigrate part of the task to other local asip and update fragment list.
- 6) If no corrspnd asip for part, immigrate whole task to other cluster.
- 7) Check fragment list to joint fragment parts.
- 8) Immigrated task in asip with lowest load to other local asip.
- 9) Update new R and S .
- 10) End while.

Fig. 5. Refine algorithm with designed heuristics

IV. EXPERIMENTAL EVALUATION

Simulation environment is shown in table III:

TABLE III. SIMULATION PARAMETER SETTING

Parameter name	Value
N_a (number of ASIP in a cluster)	20
N_c (number of cluster)	20
ORX, OTX and DSP core number (N_{core}) per ASIP	1, 1, 10
5MHz V-BS number	300 (low pressure), 600 (high pressure)
10MHz V-BS number	150 (low pressure), 300 (high pressure)
20MHz V-BS number	100 (low pressure), 200 (high pressure)
Load model of single DAG	Continuous random wave
S_{guard} for element A and B in S	0.95
Peak of a in R	0.25(5M), 0.45(10M), 0.85(20M)
Peak of b in R	0.25(5M), 0.45(10M), 0.85(20M)
Peak of c in R	$1/N_{core}(TX)$, $1/N_{core}(RX_5M)$, $2/N_{core}(RX_10M)$, $3/N_{core}(RX_20M)$,

This experiment is done in both high pressure and low pressure modes. The variation of the idle ASIP in one cluster is recorded in Fig.6. Approximately 10% of ASIPs can be power off in high pressure mode and approximately 45% in low pressure mode.

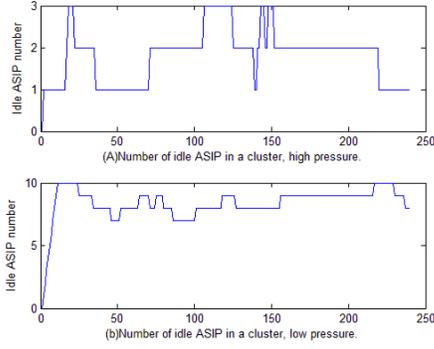


Fig. 6. Idle ASIP number in one cluster under different pressure mode

The hardware occupying ratio can be simply defined in (5). N_i is the number of actually used ASIPs of i_{th} cluster in the proposed algorithm, and $N_{fix_mapping}$ is the number of the ASIP need in fix mapping method.

$$R_u = \frac{\sum_{i=1}^{N_{cluster}} N_i}{N_{fix_mapping}} \quad (5)$$

The value of the R_u is shown in Fig.7. It can be found that the hardware occupying ratios in two different modes are nearly the same, which is approximately 70%~75%. This result illustrates that only 70%~75% hardware resources are required by proposed method when compared to the fix mapping method. This result is based on the random load model with assuming that the expected total load of all the V-BSs is stable.

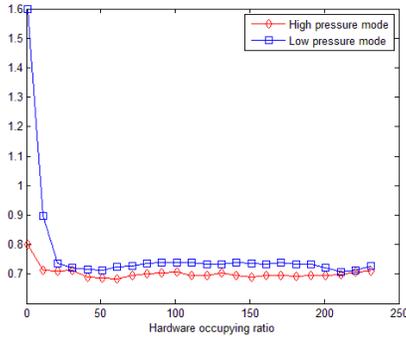


Fig. 7. Hardware occupying ratio under different pressure mode

V. CONCLUSION

The target of LTE PHY layer virtualization is to provide a better way to organize hardware resources. Therefore, how to represent the PHY layer task load and how to schedule the tasks efficiently are mainly concerned and discussed in this paper. A four step method called *Increment Scheduling with Pre-defined Parallelism* is proposed to reduce hardware resources consumptions. According to the experiment results, both consolidation and load balance features are realized in the

algorithm and only 70%~75% ASIPs are need when compared to the fixed mapping method.

ACKNOWLEDGMENT

This work was supported by the key project of the National Natural Science Foundation of China (No. 61431001), National High-Tech R&D Program (863 Program 2015AA01A705) and New Technology Star Plan of Beijing (xx2013052).

REFERENCES

- [1] R.P. Goldberg, "Survey of Virtual Machine Research," Computer, June 1974, pp. 34-45.
- [2] R. Uhlig, G. Neiger, D. Rodgers, A. L. Santoni, F. C. Martins, A. V. Anderson, S. M. Bennett, A. Kagi, F. H. Leung, and L. Smith, "Intel virtualization technology," IEEE Comput., vol. 38, no. 5, pp. 48-56, May 2005.
- [3] Yong Hua Lin, Ling Shao, Zhenbo Zhu, Qing Wang, Ravie K. Sabhikhi, "Wireless network cloud: Architecture and system requirements" in IBMRD, vol. 54, no. 1, pp. 4-12, 2010
- [4] Jingchu Liu; Tao Zhao; Sheng Zhou; Yu Cheng; Zhisheng Niu, "CONCERT: a cloud-based architecture for next-generation cellular systems", Wireless Communications, IEEE, Year: 2014, Volume: 21, Issue: 6, Pages: 14 - 22.
- [5] Wubben, D.; Rost, P.; Bartelt, J.S.; Lalam, M.; Savin, V.; Gorgoglione, M.; Dekorsy, A.; Fettweis, G., "Benefits and Impact of Cloud Computing on 5G Signal Processing: Flexible centralization through cloud-RAN," Signal Processing Magazine, IEEE, Year: 2014, Volume: 31, Issue: 6, Pages: 35 - 44.
- [6] Manli Qian, Yuan Yuan Wang, Yiqing Zhou and Jinglin Shi, "A super base station based centralized network architecture for 5G mobile communication", Digital Communications and Networks 03/2015; 54. DOI: 10.1016/j.dcan.2015.02.003
- [7] Li Guangjie; Zhang Senjie; Yang Xuebin; Liao Fanglan; Ngai Tin-fook; Zhang, S.; Kuilin Chen, "Architecture of GPP based, scalable, large-scale C-RAN BBU pool", Globecom Workshops (GC Wkshps), 2012 IEEE Year: 2012, Pages: 267 - 272.
- [8] ZhenBo Zhu, Parul Gupta, Qing Wang, Shivkumar Kalyanaram, Yonghua Lin, Hubertus Franke, Smruti Sarangi, "Virtual base station pool: towards a wireless network cloud for radio access networks", Proceedings of the 8th ACM International Conference on Computing Frontiers. ACM, 2011: 34.
- [9] Maxime Pelcat, Jean-Francois Nezan, Slaheddine Aridhi. "Adaptive Multicore Scheduling for the LTE Uplink", AHS, 2010.
- [10] Manjunath Kudlur, Scott A. Mahlke, "Orchestrating the execution of stream programs on multicore platforms", PLDI, pp. 114-124, 2008.
- [11] Paul M. Carpenter, Alex Ramirez, Eduard Ayguadé, "Mapping stream programs onto heterogeneous multiprocessor systems", CASES, pp. 57-66, 2009.
- [12] H. Topcuoglu, S. Hariri, and M.-Y. Wu. "Task scheduling algorithms for heterogeneous processors". In IPPS/SPDP Workshop on Heterogeneous Computing, pages 3-14, San Juan, Puerto Rico, Apr. 1999.
- [13] Topcuoglu, H., Comput. Eng. Dept., Marmara Univ., Istanbul, Turkey, Hariri S., Min-You Wu, "Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing",
- [14] Shan Huang, Ziyuan Zhu, Yongtao Su, Jinglin Shi. "A system-level design approach for SDR-based MPSoC in LTE baseband processing". MWSCAS, pages: 623 - 626, 2014.
- [15] Tang Shan, Zhu Ziyuan, Su Yongtao, "System-level design methodology enabling fast development of baseband MP-SoC for 4G small cell base station", Design, Automation and Test in Europe Conference and Exhibition-DATE, pp. 1-6, 2014.