Package Assignment and Processing Resource Allocation for Virtual Machines in C-RAN

Alireza Zamani, Saeed Shojaee, Rudolf Mathar and Anke Schmeink

Institute for Theoretical Information Technology
RWTH Aachen University, D-52074 Aachen, Germany
Email:{zamani, shojaee, mathar, schmeink}@ti.rwth-aachen.de

***Abstract*—Cloud-Radio Access Networks (C-RANs) are known for their potential to accommodate the heavy processing required by the exponentially increasing data traffic. Thanks to virtualization techniques, the baseband processing in C-RANs, can now be performed on virtual machines (VMs) at the central unit (CU). This is a priceless tool for improving the processing efficiency, as it provides the opportunity for dynamic allocation and sharing of processing resources. In this work, we investigate the assignment of processing jobs and the allocation of resources among the virtual machines, while reducing the overall power consumption of the VMs. Furthermore, the developed model also accounts for communication overhead, which may incur due to package dependencies. The performance of the proposed technique is investigated with varying system constraints and the obtained results demonstrate the potential power savings made possible using the proposed method.**

I. Introduction

C-RAN continues to emerge as a strong contender for meeting the demands imposed on future network generations. The concept aims to achieve performance gains by decoupling the baseband processing units (BBUs) from the remote radio heads (RRHs). The BBUs are then pooled together at the CU where, the baseband processing is carried out by VMs, which require processing resources, e.g., processing speed and memory capacity. Recent works, [1] and [2], have studied the benefits and challenges of C-RAN, highlighting the reductions in expenditure made possible via centralized management of radio and processing resources. The centralization and scalability of C-RAN also provide the necessary ingredients for the implementation of cooperative techniques, e.g., [3] and [4], which offer substantial performance gains at the cost of computational complexity. However, in order for C-RAN to become a commercial success, many challenges are yet to be resolved. On this note, the assignment of processing jobs and computation of the optimal number of VMs, as well as their required processing resources, represents an interesting problem which has not been adequately studied. This is particularly challenging, since due to pooling of the resources, allocation of any resource impacts the amount of available resources to other VMs. This problem is also of inherent value as VM utilization is directly related to processing power consumption at the CU. Moreover, considering that baseband processing is responsible for a considerable portion of the overall power expenditure of a network, minimizing the power consumption is of substantial interest. In this work we provide an approach for joint job assignment and processing resource allocation, while minimizing the power consumption of VMs in the BBU pool. A mathematical framework is developed, which jointly determines the optimal number of VMs, their processing resources and the package allocation among them, under various system and VM constraints.

In the existing literature, the authors in [5] and [6] consider a C-RAN network and aim to minimize the number of active BBUs, while satisfying traffic load demands. However, the allocation of processing resources or the power consumption of the VMs in the BBU pool is not investigated. The work in [7] proposes a two stage approach, which first allocates the physical resource blocks to users and then determines the number of BBUs required to satisfy the demand. The latter is formulated as a knapsack problem, which assigns RRHs to BBUs and does not take into consideration the power consumption of the BBUs nor determine their processing resources. Lastly, the authors in [8] consider a multi-cloud scenario and implement heuristic algorithms to select a data-center and it's processing resources, while minimizing the path length of the packets as a way of reducing the delay. In general, the processing resource allocation, is often oversimplified to a knapsack problem, which is an unrealistic model, considering that the behaviour of processing speed is different to memory. Additionally, the utilization of resources is strongly linked with the power consumption of the VMs and hence the overall operational expenditure of the system, which has not been well studied. Lastly, further complications rise when considering more realistic models which include potential package inter-dependency, where the processing of one package requires data from another. To the best of our knowledge, no current work considers joint package assignment and allocation of

(multiple) resources with such a detailed system model, while also minimizing the overall VM power consumption.

Paper Organization: In Section II, the system model is provided, detailing the typical constraints and limitations in a BBU pool. The corresponding optimization problem is described in Section III along with its convex reformulation. Section IV includes the simulation setup and discussion on the performance of the proposed model in comparison to conventional techniques. Lastly, the conclusions of our work are provided in Section V.

II. System Model

The general architecture of a C-RAN is depicted in Fig. 1. Note that although the incoming traffic typically goes through a queuing system before being assigned, we are mainly interested in finding the optimal assignment of packages and allocation of resources in the BBU pool.

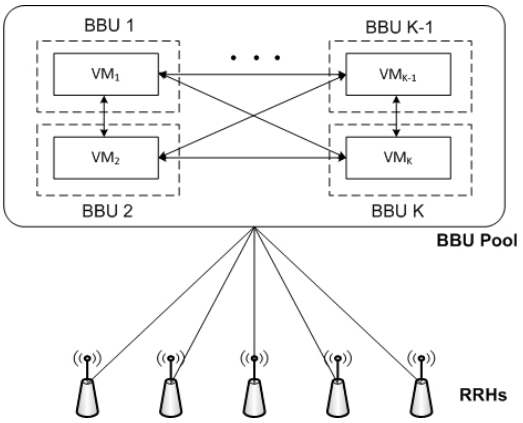

Fig. 1: Schematic of a typical C-RAN and its BBU pool.

In this work, the assignment of N packages (processing jobs) to the VMs in the BBU pool is considered. The size and overall delay requirement of the i-th package is indicated by γ_i and δ_i, respectively. The assignment of the i-th package to the j-th VM is described by the binary variable α_{ij}. Instead of defining the number of required VMs as a single integer, we introduce a set of VMs, $\{VM_1, VM_2, ..., VM_K\}$, where K is the maximum number of VMs the BBU pool can support. This is an important step as it allows us to define a binary switch indicator ν_j, to describe the corresponding VM as active or inactive and consequently obtain a suitable mathematical structure for an off-the-shelf numerical solver. There exists a limit on the total physical memory available in the BBU pool, which can be allocated to active VMs, denoted by B, as well as individual constraints on the maximum processing speed and memory which can be allocated to a given VM, shown by ϕ_{max} and Ψ_{max}, respectively. Moreover, the allocated processing speed and memory capacity of the j-th VM are defined using the variables f_j and m_j, correspondingly. A detailed description of the power consumption model of a VM is provided later, where the idle power consumption is given by P_{on} and the maximum power consumption is denoted by P_{max}. Furthermore, we assume the power consumption is zero when the VM is off. The interdependency of the i-th package to the l-th package, denoted by c_{il}, is given as a fraction of the size of the i-th package itself, in the range of $[0, 1]$. This essentially describes what percentage of the total package size is dependant to another package. It is worth mentioning that the communication overhead, is assumed to only add to the memory requirement of the VM. This is justified by the fact that the already processed bits only need to be exchanged and not reprocessed. Subsequently, the interconnecting link between the VMs, which carries the communication overhead, has a limited capacity denoted by z_j. Note that for the sake of simplicity we assume that the links are bi-directional and that z_j is the sum capacity of all the links connected to the j-th VM.

A. Assignment & Switch Indicators

The assignment of the i-th package to the j-th VM is described as a binary variable. Furthermore, all packages are required to be assigned and no package can be assigned to two VMs. This can be formulated as the constraint below

$$\alpha_{ij} \in \{0, 1\}, \quad \forall i, j, \tag{1}$$

$$\sum_j \alpha_{ij} = 1, \quad \forall i. \tag{2}$$

The number of required VMs is given by the sum of a binary switch indicator, based on the arguments provided earlier, and is limited by the maximum number VMs that the BBU supports. This can be subsequently written as

$$\nu_j \in \{0, 1\}, \quad \forall j, \tag{3}$$

$$\sum_j \nu_j \leqslant K. \tag{4}$$

Note that obtaining a similar structure would not have been possible if the number of required VMs was instead formulated as an integer. It is important to note that there exists a direct relationship between the assignment variable and the VM switch, where a VM should only be active if a package is assigned to it, this can be formulated by the following constraint

$$\nu_j \geq \alpha_{ij}, \quad \forall j. \tag{5}$$

B. VM Constraints

The allocated memory resource to a VM should accommodate the total memory requirement of the packages assigned to the j-th VM as well as the communication overhead. Additionally the total memory in the

BBU pool available for allocation is limited, while there also exists a restriction on the maximum memory that can be allocated to an individual VM. These restrictions are, correspondingly, formulated as

$$\sum_i \alpha_{ij}\gamma_i + \sum_{i,l} c_{il}\gamma_i|\alpha_{ij} - \alpha_{lj}| \leqslant m_j, \quad \forall j, \tag{6}$$

$$\sum_j m_j \leqslant B, \tag{7}$$

$$m_j \leqslant \Psi_{max}, \quad \forall j. \tag{8}$$

The allocated processing speed to a given VM also has an upper-bound and may be shown as

$$f_j \leqslant \phi_{max}, \quad \forall j. \tag{9}$$

Furthermore, it is required for the overhead communication for the j-th VM to not exceed the capacity of the interconnection link, this may be formulated as

$$\sum_{i,l} c_{il}\gamma_i|\alpha_{ij} - \alpha_{lj}| \leqslant z_j, \quad \forall j. \tag{10}$$

Note that for the sake of simplicity, z_j is defined as the sum capacity of all the interconnection links of the j-th VM, nonetheless, the model can be easily extended.

C. Delay Requirement

The overall delay experienced by a package is broken down to queuing time and processing time, as shown below

$$\tau_{i,q} + \alpha_{ij}\tau_{j,p} \leqslant \delta_i, \quad \forall i, j \tag{11}$$

As the scheduling of the packages lies outside the scope of our work, the queuing delay is assumed to be constant and given per package. The processing delay of a package, however, depends on the processing speed and the overall load on the VM which the package is assigned to. This can be defined in the following way

$$\tau_{j,p} = \frac{\sum\limits_l \alpha_{lj}\gamma_l}{f_j}, \quad \forall j \tag{12}$$

Note that this formulation expresses the processing delay as a function of the overall load (assigned packages) and the allocated processing speed, which is a more technically sound model, in contrast to works in literature where the processing speed is assumed to be a simple block resource like memory.

D. VM Power Consumption Model

The VM power model used in our work follows [9], where the relationship between VMs power consumption and the processor utilization, denoted by u_j, is described below

$$P_j(u_j) = \begin{cases} 0 & \text{if } u_j = 0 \\ P_{on} + (P_{max} - P_{on})u_j & \text{if } u_j \neq 0 \end{cases} \tag{13}$$

where u_j is considered to be the utilization of the VM in terms of processing speed, given as $u_j = f_j/\phi_{max}$. Since we have defined a VM switch indicator in our system model, a more suitable (and concise) reformulation of the power consumption can be obtained as follows

$$P_j(u_j) = P_{on}.\nu_j + (P_{max} - P_{on})u_j, \quad \forall j \tag{14}$$

III. Optimization Problem

With the aforementioned system model definitions and restrictions, the power consumption minimization is described by the following optimization problem

$$\min_{\alpha_{ij},\nu_j f_j, m_j} \quad \sum_j P_j(u_j)$$
$$\text{s.t.} \quad (1),(2),(3),(4),(5),(6),(7),(8),(9),(10),$$
$$(11),(14)$$

Unfortunately, the multiplication of the assignment variable by itself in constraint (11), $\alpha_{ij}\tau_{j,p} = \alpha_{ij}\frac{\sum_l \alpha_{lj}\gamma_l}{f_j}$, violates the linearity of the problem.

A. Linearization

However, as the multiplication of two binary variables resembles an *AND* operation, we show that linearization of the constraint is possible, as shown in [10], by introducing a new binary auxiliary variable, β_{ilj}, of higher dimension subject to the following constraints

$$\beta_{ilj} \leq \alpha_{ij}, \quad \forall i, l, j \tag{15a}$$
$$\beta_{ilj} \leq \alpha_{lj}, \quad \forall i, l, j \tag{15b}$$
$$\beta_{ilj} \geq \alpha_{ij} + \alpha_{lj} - 1, \quad \forall i, l, j \tag{15c}$$

using the linearization technique above, the following reformulation of constraint (11) holds

$$\alpha_{ij}\tau_{i,p} = \frac{\sum\limits_l \beta_{ilj}\gamma_l}{f_j}, \quad \forall i \tag{16}$$

Consequently, the objective and all the constraints become linear, allowing the problem to be cast as the Mixed integer linear programming (MILP) optimization problem compatible with Gurobi [11] as shown below

$$\min_{\alpha_{ij},\nu_j f_j, m_j} \quad \sum_j P_j(u_j)$$
$$\text{s.t.} \quad (1),(2),(3),(4),(5),(6),(7),(8),(9),(10),$$
$$(11),(14),(15a),(15b),(15c),(16)$$

IV. Simulation Results & Discussion

In this section, the performance of our joint assignment and resource allocation method is evaluated. In order to obtain a better insight, the performance of the proposed approach is compared with the following common techniques:

1) Greedy Allocation: here, packages are assigned to a VM until it is completely full or any constraint is violated. A greedy allocation of resources is also considered, where the maximum amount of available resources is allocated to the utilized VMs. This scheme depicts a case with the least number of VMs albeit with high utilization, however, it may also experience package loss, i.e., when assigning a package violates a constraint.
2) Equal Allocation: this strategy distributes the packages and resources equally among VMs. Thereby utilizing the maximum number of VMs. Note that equal allocation can also result in package loss due to insufficient resources available at the VM.
3) Random Allocation: here the packages are randomly assigned to different VMs and the resources are allocated based on the assignment. This is essentially a two stage scheme; with the assignment of packages in the first stage and the computation of required resources in the second. Note that if the necessary resources for the current assignment violate any constraints the package will be dropped.

The simulation parameters used generally follow the literature [9], with the exception of processing resource constraints, which were set to investigate the full range of system performance, nonetheless they can be changed as per specification. The complete set of simulation parameters are provided in Table I. The overall power consumption of the VMs in the BBU pool are investigated with varying values of the package size and delay. Note that these were chosen due to their direct affect on required memory and processing speed, respectively. It is worth mentioning that in our studies 12 packages were considered. While for the simulations involving random variables, an average was taken over 100 realizations. Since our preliminary inspections indicated that, when package dependency is included, the average packages loss in the conventional methods drops too low to acquire any meaningful comparison to our proposed model, henceforth we assume no package dependency. Nonetheless, it is worth mentioning that the our proposed model is perfectly capable of supporting package dependency and VM to VM overhead communication.

TABLE I: Simulation Parameters

Parameter	**Settings**
Maximum Processor Power, P_{max}	276 watts
Idle Processor Power, P_{on}	109 watts
Maximum Number of VMs, K	5
Total Available Memory, B	150 Mb
VM Maximum Memory, Ψ_j	75 Mb
VM Maximum Processing Speed, ϕ_j	50 Gbps

The power consumption performance is studied with increasing package sizes in Fig. 2. It is observed that the proposed method achieves significant reductions,

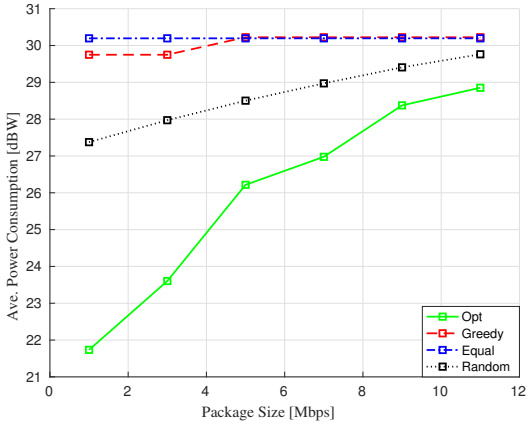

Fig. 2: Power consumption vs. package size. The proposed method consistently outperforms conventional techniques in terms of power consumption.

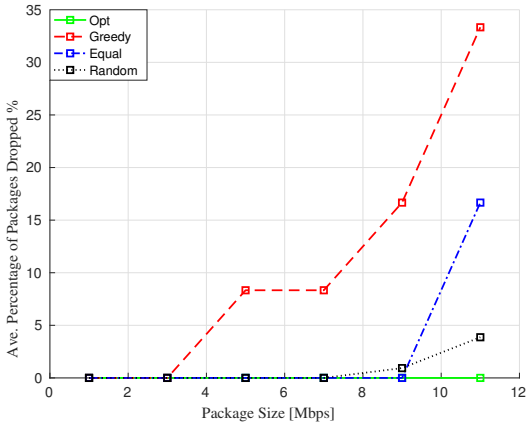

Fig. 3: Package drop vs. package size. The package loss in other methods grows exponentially with the package size, compared to the proposed method.

especially with smaller package sizes. It is interesting to observe that equal and greedy allocation, which represent two extremes for the number of active VMs (highest and lowest, respectively), both incur a higher power consumption. This highlights the importance and potential performance gains in jointly optimizing the number of active VMs, their resource allocation and the package assignment. It is also evident that even under heavy loads, the proposed method still consumes less power in comparison to the other techniques.

For further insight, the average percentage of packages dropped in each method was also investigated. Figure 3 illustrates that the greedy approach results in the highest package loss. By studying Fig. 2 and 3 together, it can be stated that, not only does the proposed approach achieve a zero percent package drop rate, due to constraint (2), but does so with less power consumption.

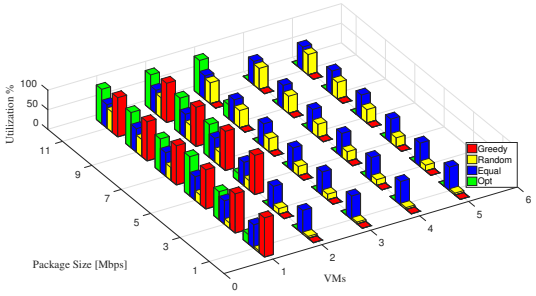

Fig. 4: VM utilization vs. package size. The joint optimization framework aims to find a balance between the number of employed VMs and their utilization.

Additionally, we investigate the utilization of VMs under the varying package sizes. Figure 4 illustrates the average distribution of packages and consequently the utilization of VMs. For instance, the flatness of the greedy algorithm in Fig. 3 for package size between 5-7 Mb can be explained by observing that during that range a second VM is employed, which is yet to be completely filled, hence the package loss remains constant for that period. Further insight can also be gained in terms of the number of VMs each method deploys. It is evident that, equal allocation uses the highest number of VMs possible, while greedy allocation uses the least.

The power consumption is also investigated in Fig. 5 with values of the processing delay tolerance ranging from 1 ms to 100ms, while assuming a fixed package size of 9Mb. Note that this represents an important study as the processing delay directly impacts the amount of processing speed resources required, which subsequently relate to VM utilization and power consumption. The obtained results generally illustrate that as the delay tolerance increases, the power consumption declines, due to the lower amount of processing speed required. It can be observed that with the greedy allocation, since VMs operate at maximum speed, the power consumption is indifferent to changes in delay tolerance.

V. Conclusion

In this paper a new technique for the assignment of packages and allocation of processing resources between the VMs in a BBU pool was introduced. The proposed method jointly determines the optimal assignment of packages, number of employed VMs and their allocated resources, while minimizing the overall power consumption. Simulation results provide evidence that the proposed method is able to consistently outperform other conventional schemes. Considering that the baseband processing power constitutes a large portion of the overall power expenditure of a network, the performance gains achieved by using the developed framework are highly valuable.

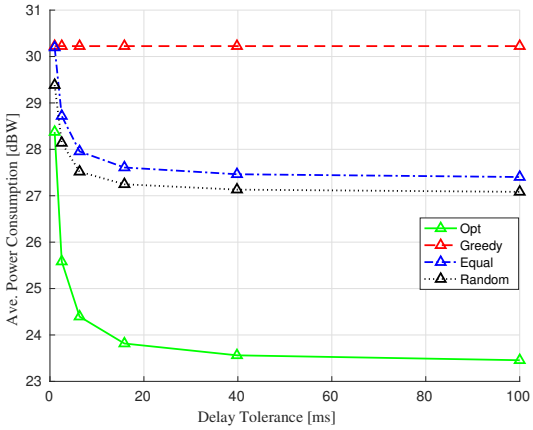

Fig. 5: Power consumption vs. processing delay tolerance. The proposed MILP framework yields significant power savings by optimal VM utilization.

VI. Acknowledgement

The project has been supported by the federal ministry of education and science (BMBF) grant 01IS17073.

References

[1] H. Dahrouj, A. Douik, O. Dhifallah, T. Y. Al-Naffouri, and M. S. Alouini, "Resource Allocation in Heterogeneous Cloud Radio Access Networks: Advances and Challenges," *IEEE Wireless Communications*, vol. 22, no. 3, pp. 66–73, June 2015.

[2] M. Peng, Y. Sun, X. Li, Z. Mao, and C. Wang, "Recent Advances in Cloud Radio Access Networks: System Architectures, Key Techniques, and Open Issues," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2282–2308, 2016.

[3] A. Zamani, S. Shojaee, O. Taghizadeh, and A. Schmeink, "Beamforming Optimization with Hybrid Association in C-RANs Under a Limited Backhaul," in *IEEE 14th International Symposium on Wireless Communication Systems (ISWCS)*, Bologna, Italy, Aug. 2017.

[4] A. Zamani, O. Taghizadeh, and A. Schmeink, "A Novel Joint Precoding and Association Optimization Framework for C-RANs with Restricted Fronthauls," in *IEEE 22nd International ITG Workshop on Smart Antennas (WSA)*, Bochum, Germany, March 2018.

[5] N. Yu, Z. Song, H. Du, H. Huang, and X. Jia, "Multi-Resource Allocation in Cloud Radio Access Networks," in *IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[6] D. Mishra, P. C. Amogh, A. Ramamurthy, A. A. Franklin, and B. R. Tamma, "Load-Aware Dynamic RRH Assignment in Cloud Radio Access Networks," in *IEEE Wireless Communications and Networking Conference*, April 2016, pp. 1–6.

[7] M. Y. Lyazidi, N. Aitsaadi, and R. Langar, "Dynamic Resource Allocation for Cloud-RAN in LTE with Real-Time BBU/RRH assignment," in *IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[8] M. Alicherry and T. V. Lakshman, "Network Aware Resource Allocation in Distributed Clouds," in *Proceedings IEEE INFOCOM*, March 2012, pp. 963–971.

[9] Q. Wu, F. Ishikawa, Q. Zhu, and Y. Xia, "Energy and Migration Cost-Aware Dynamic Virtual Machine Consolidation in Heterogeneous Cloud Datacenters," *IEEE Transactions on Services Computing*, vol. PP, no. 99, pp. 1–1, 2017.

[10] H. P. Williams, *Model Building in Mathematical Programming, 5th Edition*. John Wiley & Sons, Inc, 2013.

[11] I. Gurobi Optimization, "Gurobi Optimizer Reference Manual," 2016. [Online]. Available: http://www.gurobi.com