Modelling the Power Consumption and Trade-offs of Virtualised Cloud Radio Access Networks

Journal:	*IET Communications*
Manuscript ID	COM-2016-1396.R1
Manuscript Type:	Research Paper
Date Submitted by the Author:	01-Feb-2017
Complete List of Authors:	alhumaima, raad Al-Raweshidy, H.; Brunel University, Wireless Communications & Networks Group
Keyword:	COMMUNICATION SYSTEMS, POWER MEASUREMENT, MODELLING

Modelling the Power Consumption and Trade-offs of Virtualised Cloud Radio Access Networks

Raad S. Alhumaima, and H. S. Al-Raweshidy

WNCC, School of Engineering, Design and physical sciences, Brunel University London, Uxbridge, Middx., UB8 3PH, UK

1234914@my.brunel.ac.uk

Abstract: In large-scale computing centres, the advancement of knowledge in regards to the predicted power consumption (PC) and concerns of host servers that run virtual machines (VMs) could improve the capacity planning and networks' Energy Efficiency (EE). In this paper, a parameterised power model is proposed to explore the individual components within the virtualisation based cloud-radio access network (vC-RAN). The model evaluates the PC and trade-offs of a server undergoing virtualisation. After, cooling and total PC for C-RAN architecture with and without virtualisation have been compared using differentiated parameters, such as varying number of bare-metal base band units (BBUs), VMs and system's resource blocks (RBs) share/bandwidth. The results show dramatic decrease in the total PC via virtualising the core network (CN). In addition, the degraded performance of each virtualised server is demonstrated via modelling the execution time and overhead costs. These costs have been resulted from increasing the number of hosted VMs and utilised RBs by each VM.

1. Introduction

Driven by the need to provide at least 10 times higher spectral and energy efficiency (EE) in 5G networks, mobile operators and equipment vendors have privileged implementing cloud radio access network (C-RAN) [1]. In C-RAN, the base band units (BBUs) are responsible for processing the upper layers, and most of the physical layer functions are shifted to a centralised location, called a BBU pool. At the cell site, the rest of physical layer functions such as radio frequency and optical to electrical conversions are tackled by the remote radio heads (RRHs). The RRHs are connected via wireless or optical fiber to the BBU pool. In contrary to existing cellular networks, C-RANs becomes exceedingly capable of implementing cooperation algorithms, dynamically utilising the available spectrum, exploiting the load variation to run fewer compute resources, and integrating with the new 5G enabling technologies. Additionally, C-RAN curtails the operational expenditures (OPEX) and capital expenditures (CAPEX) due to reduced maintenance costs, less site visits and leases. Despite that such planning can unleash the network potentials to the maximum regarding EE and cost of operation. However, intensifying the number of deployed RRHs and active BBUs can maintain a considerable amount of power consumption (PC). Recently, the research community embraced the use of network function virtualisation (NFV) techniques in the cloud for several reasons: flexible allocations for the available network resources, enabling automation in the servers' operation and configuration, reducing the cost of maintenance, supporting

multi-tenancy services, and potential reduction in the energy consumption. By using NFV, running fewer servers in the pool becomes possible while fulfilling the users' (UEs) quality of service requirements [1], [2], [3], [4]. Therefore, NFV was devoted to provide significant increase in the EE, which allows service providers to execute network's functions using software rather than running proprietary built or dedicated appliances. Due to the latter, updating and expressing new services and applications that are necessary to enable 5G is increasingly intractable. NFV also enables the use of general off-the-shelf equipment to run these functions, or called virtual machines (VMs). However, running multiple VMs on a single hardware requires a supervisor or manager. This manager is also called a hyper-visor (HV). The HV is a software that runs on the host's hardware to dynamically control and allow the host server to be shared by guest VMs. Each VM then appears to hold or utilise server's memory (RAM), processors (CPU), network interface card (NIC) and hard drive (HDD) all to itself. But in fact, each VM shares these resources with other VMs. The HV then assures that the hosted VMs cannot obstruct each other while accessing these resources. The presence of the HV within the host server increases RAM accesses, CPU computations and storage usage. This increment happens as a consequence to the increased interruptions and orchestration sessions between the VMs and HV. Consequently, the VMs have yielded extra overhead added to the server PC. Such matter urges to evaluate this increment in the PC and other trade-offs, such as the execution time delay within each host compared to the non virtualised servers. Furthermore, it is necessary to compare the entire PC of the core network (CN) or BBU pool with and without using virtualisation [5]. In general, virtualisation greatly reduces the overall network's PC by using fewer servers. However, such gain requires awareness to the consequences facing each virtualised server regarding the PC and latency, as follows:

1.1. Advantages and Disadvantages of NFV

1. NFV appears as a compensator to the increased PC due to integrating new network services and enablers such as software defined networks (SDN) and load balancers appliances with C-RAN [6]. This can be achieved by provisioning and sharing for the available servers' resources while cascading multiple VMs to be run by fewer servers. Each server then holds several VMs, and each VM performs different network's function.

2. A virtualised server with 1 VM may take about 5 times more delay to process a packet compared to bare metal counterpart [5]. This delay comes from the fact that each VM owns a small share of the existing server resources to compute its load. This means when there are no resources available at any time, the VM must queue its load and wait for a window to be opened again by the HV. This matter urges to provide optimisation techniques to enhance the HV's cycle scheduling. By which, the fast and dynamic operation can relief such constraint.

3. The virtualised server itself gains a PC overhead due to fully utilising its resources by the VMs. In addition, the PC of each VM is directly proportional to the allocated bandwidth or resource block (RBs), which increases the dynamic consumption of the server to the maximum.

4. While the number of VMs increases in one server, this reduces the resources' shares allocated to each VM, which more increases the delay to process UEs' traffic [7]. Therefore, optimising the number of the VMs installed in one host server is prerequisite in regards to the traffic volume demand and available server resources. Consequently, the VMs can constantly meet their real time requirements.

Note that, PC probably stands for power consumed or consumption.

1.2. Related Work

There have been several researches aim to investigate modelling the network's PC on components and system level. Herein, merely the system level PC models are deliberated upon, as it is the focal point within this study. In [8] and [9], parameterised PC models are presented by Energy Aware Radio and network TecHnologies (EARTH) PC evaluation project. These have been used to linearise the PC of the SotA BSs (i.e., Macro, Micro, Pico, etc.). In which, the PC of the BS is directly proportional to the allocated bandwidth. However, such models were unable to estimate the amount of PC in the futuristic and hybrid networks, such as C-RAN, Heterogeneous C-RAN, SDN based C-RAN or virtualised networks. In [10], the investigation concerning the level of power reduction that can be attained if C-RAN is deployed instead of the traditional BSs deployment. In [6] SDN based C-RAN PC model is evaluated. Moreover, there have been several proposals, algorithms and paradigms to adapt and implement NFV in the cloud based networks [11]. For example: NeFuCloud, PLayer, CloudNaaS, APLOMB, PACE, SIMPLE, CloudNFV and REALTIME CLOUD. These works investigate and define different mechanisms associated to the various modes of operations, and mostly they rely on open source tools to implement the suggested paradigms. Unfortunately, they lack for the mathematical representation, in-depth analysis and evaluation on the matter of power cost and allocations. In [5], the virtualisation effect upon the PC of a single server while running certain packages and applications is experimentally tested. This work however contemplate in association to a single server case study without providing a mathematical model based platform to measure the PC for components or system level, similarly in [12] and [13]. In [14], the work has provided a comprehensive survey for the available PC models including virtualised, non virtualised servers or data centers. These models have been classified as intrusive, machine learning and software based models according to the approach used to measure the PC. Generally, these models are expected to add additional PC to the measured PC. Intrusion based models require to install intrusive tools and events counters which make the PC measuring expensive and complex. Software based models require additional application or separate device to run, this method is also a power consuming and complex. Machine learning algorithms are based on heuristics, this method however is time and power consuming. These algorithms may require additional device to run. On the other hand, our model is much simpler and costless, it is only based on the number of hosted VMs, allocated bandwidth/RBs to each VM and components' data sheets.

1.3. Main Contributions

Hardware virtualization adds substantial overhead, as a busy web-server consumes about 40% more power, synchronised with less efficiency when compared to a non-virtualized server [5]. Therefore, to decide whether or not it is worthwhile to compromise the network performance with total PC reduction, the following contributions have been accomplished:

1. Modelling the way active VMs and processed RBs by each VM increasingly affect the PC of the host server in terms of CPU, RAM, NIC and HDD. Based on such parameters, the model would provide a realistic, accurate and easy visualisation to the PC modelling compared to intrusive, software and utilisation level based models. Furthermore, the HV's PC is modelled.
2. Comparing cooling and the total PC of BBU pool and CN with and without virtualisation

by considering different parameters, such as number of VMs, BBUs and RBs. Besides, modelling the PC of CN's control units, i.e., mobility management entity (MME), serving gateway (SGW) and packet gateway (PGW).

3. The latency, which is concurrent with increasing the number of VMs and executed RBs at each VM is also modelled. This modelling gives a prediction to the deficiency expected in the virtualised networks.

2. Server Power Consumption

Server's PC is initiated from the higher computation levels, generated I/O instructions' executions and compound accessing for the device resources by the aggregated VMs' applications [7]. Generally, there are four major participants involved within the constituency of a server's PC, these are: RAM, CPU, NIC and storage or HDD, the PC model of each virtualised part can be expressed as follows:

2.1. RAM Power Model

Usually, each VM requires a share from the RAM to utilise during operation. However, when the number of installed VMs (N) increases, the size of the RAM as well as its corresponding PC intensifies as a result, in order to handle an amplified amount of RAM usage requests [15]. Therefore, it is compulsory to identify the appropriate RAM size (Z_{RAM}) to place in a server that contains N VMs. The proposed method initially describes the change in the RAM size corresponding to the change in N, i.e., $\frac{dZ_{RAM}}{dN} = \alpha Z_{RAM}$. Solving this equation yields $Z_{RAM}(N) = Z_{int}e^{\alpha N}$, where Z_{int} is the initial RAM size. Afterwards, the corresponding PC ($P_{RAM}(Z_{RAM})$) of the RAM can be modelled based on recognizing the maximum and initial PC of the RAM size. In this case, another constant ($\beta << 0.1$) is introduced to shape such change in the PC:

$$\frac{dP_{RAM}}{dZ_{RAM}} = \beta P_{RAM} \tag{1}$$

When solving equation (1), it yields:

$$P_{RAM}(Z_{RAM}) = P_{intRAM}\, e^{\beta Z_{RAM}} \tag{2}$$

Where P_{intRAM} is the initial PC of the RAM. It is worth noting that the constants α and β may vary based on type of the RAM used and its initial and maximum PC.

Alternatively, a straightforward relation between N and the RAM PC is modelled by representing RAM size as $Z = 2^{i+1}GB$, where i is an index ($i = 0, 1, 2, 3...$) to shape the commercial RAM sizes $i.e., Z = (2, 4, 8, 16, 32GB, ...)$. After, if P^i_{RAM} is the PC of the RAM which holds i index, then the change in PC of this specific RAM dP^i_{RAM} is correlated to the change of N (dN). To estimate this change, the constant ν is presented, as follows:

$$\frac{dP^i_{RAM}}{dN} = \nu P^i_{RAM} \tag{3}$$

When solving equation (3), it produces:

$$P^i_{RAM}(N) = P^i_{intram}\, e^{\nu N} \tag{4}$$

Where P^i_{intram} is the initial PC of RAM size with index i.

2.2. CPU Power Model

The initial assumption for modelling the PC of CPU is based on practical considerations that each CPU core can hold at least one VM. In other words, the number of cores per CPU (C) is always larger than the number of hosted VMs in one core (N_C), i.e., ($C \geq N_C + 1$). Subsequently, any additional VMs installed in each core means greater amount of power the core will consume until it reaches its maximum. Such affiliation is recognised as being exponential in a core PC level. Logically, the core PC (P_{core}) upsurges from an initial ($P_{intcore}$) while increasing N_C. The above assumptions can be translated into the following model:

$$\frac{dP_{core}}{dN_C} = \varepsilon P_{core} \tag{5}$$

Where ($\varepsilon < 0.1$) is a positive constant used to describe how the power is increased. Such linkage tends to be linear when ε approaches 0. However, when solving equation (5), it yields:

$$P_{core}(N_C) = P_{intcore}\, e^{\varepsilon N_C} \tag{6}$$

If P_{core} is calculated, the CPU's PC (P_{CPU}) can also be known by gathering the PCs of all cores:

$$P_{CPU} = \sum_{c=1}^{C} P_{core(c)} \tag{7}$$

Additionally, the total PC, in W, of CPUs per server (P_{server}^{CPU}) can be obtained as follows:

$$P_{server}^{CPU} = \sum_{k=1}^{K} P_{CPU(k)} \tag{8}$$

Where K denotes the total number of CPUs per server.

2.3. NIC Power Model

As each NIC is shared amongst multiple VMs simultaneously, the higher amount of VMs found per server, the more NICs are required to serve them. The PC of each NIC is obligated to an augmentation in the PC [16]. It was assumed that the maximum number of NICs (L) that can be placed in one server is shared amongst all the VMs, i.e., ($L \propto N$), where ($L \leq N$). For practical consideration, L is equivalent to 8, then each NIC (l) is assigned (N/L) VMs. Moreover, the PC of each NIC rises when the N/L is increased; as more packets will be received and transmitted via a particular NIC. Hence, the model has to outline an increment in the PC so as the virtualised NIC is driven to reach its maximum PC (P_{nic}), up from an initial value (P_{intnic}). Therefore, P_{nic} can be expressed as ($P_{nic}(\frac{N}{L})$), a function of (N/L). This linkage refers to the upsurge of P_{nic} in correspondence to the change of $\frac{N}{L}$, such behaviour can be modelled as follows:

$$\frac{dP_{nic}}{d\frac{N}{L}} = \gamma P_{nic} \tag{9}$$

When solving (9), the following solution can be obtained:

$$P_{nic}(\frac{N}{L}) = P_{intnic} e^{\gamma \frac{N}{L}} \tag{10}$$

Where γ is a constant factor. Next, the total PC, in W, of all NICs per server (P^{NIC}_{server}) can be calculated:

$$P^{NIC}_{server} = \sum_{l}^{L} P_{nic(l)} \tag{11}$$

2.4. Storage (HDD) Power Model

As a matter of fact, over-utilising the storage by hosting more VMs increases server's PC [17]. There are two main behaviours the model has to reflect on: (i) the storage is shared amongst other servers in the CN, and (ii) the VMs per server boost the storage PC; as they increase its data accesses. In other words, there will be two influencing parameters synchronised with such matter: the time (t) at which the storage is being utilised, and the number of VMs (N). Both variables then will indicate how much storage PC ($P_{storage}(t, N)$) is drawn. As far as the first variable (t) is concerned, it is considered that other servers can add, access and delete data from the tagged server. In this case, it was assumed that the storage capacity varies during time (t) to represent the sharing capability amongst CN or BBU pool's servers. The second variable in turn clarifies that N VMs increases the storage PC as a consequence to increasing the rate of accessing the stored data. When considering the time, two scenarios are determined: the first is when the storage PC ($P^{i}_{storage}$) increases by time, following the exponential method, the PC can be formulated as:

$$\frac{dP^{i}_{storage}}{dt} = -\delta P^{i}_{storage} \tag{12}$$

Where ($\delta < 1$) is a positive constant, which can control the maximum value of the storage PC, when solving (12), it yields:

$$P^{i}_{storage}(t) = P_o e^{-\delta t} \tag{13}$$

Where P_o is the initial storage PC. The second case is when the storage PC ($P^{d}_{storage}(t)$) decreases by time, following the same procedure as in (12), the PC model can be expressed as follows:

$$P^{d}_{storage}(t) = 2P_o - P_o e^{-\delta t} = P_o[2 - e^{-\delta t}] \tag{14}$$

The amount ($2P_o$) is added to uplift the initial consumption of ($P^{d}_{storage}(t)$) to start from (P_o) at time ($t = 0$), so as $P^{d}_{storage}(t)$ and $P^{i}_{storage}(t)$ start from the same point. The second variable of the storage is modelled when the PC is proportionally increased with the number of VMs. The procedure of (12) is followed and the resulting PC model can be obtained as follows:

$$P_{storage}(N) = P_o e^{\xi N} \tag{15}$$

Where ξ is a positive constant. Note that the lesser value of the constants ξ and δ are assigned, the more the model approaches to be linear. As the variables t and N are independent of each other, the model separates their PC, yet the result of each variable is aggregated, i.e., $P_{storage}(t)$ will be added to $P_{storage}(N)$. The total virtualised storage PC, in W, can be finalised as:

$$P_{storage}(t, N) = \begin{cases} P_o(e^{-\delta t} + e^{\xi N}) & \text{if } \delta \text{ increases} \\ P_o(2 - e^{-\delta t} + e^{\xi N}) & \text{if } \delta \text{ decreases} \end{cases} \tag{16}$$

2.5. HyperVisor (HV) Power model

The HV assigns a number of accesses tasks (AS) to enable the VMs to compute their load within the server's resources, if the PC per task per VM is PAS, then the PC of the HV can be modelled as follows:

$$P_{HV} = \sum_{n=1}^{N} \sum_{as=1}^{AS} PAS_{(as,n)} \tag{17}$$

Where $PAS_{(as,n)}$ is the PC of the as-th job allocated to n-th VM .

3. Components Power Model

The participating PC modules of vC-RAN paradigm encompasses mainly two parts: virtual BBU pool (vBBU pool) and virtual control plane units (i.e. vMME, vSGW and vPGW).

The vBBU server PC (P_{server}^{BBU}) consists of whatever virtualised components found in the server, this can be evaluated in W, as follows:

$$P_{server}^{BBU} = \sum_{s1=1}^{S1} [P_{RAM} + P_{server}^{NIC} + P_{BBU}^{CPU} + P_{storage} + P_{HV}]_{s1} \tag{18}$$

Where

$$P_{BBU}^{CPU} = \sum_{k1=1}^{K1} \sum_{c1=1}^{C1} P_{core(k1,c1)}^{BBU} \tag{19}$$

and $S1$, is the total number of BBU servers.

Comparably to BBU functions, PGW, MME, and SGW cores PC can be modelled following the same style by taking into consideration the functions set of each unit [6]. Therefore, the PC of control plane server (P_{server}^{cl}) is introduced as:

$$P_{server}^{cl} = \sum_{s2=1}^{S2} [P_{RAM} + P_{server}^{NIC} + P_{cl}^{CPU} + P_{storage} + P_{HV}]_{s2} \tag{20}$$

Where $S2$ denotes the total number of servers that host the control plane's servers, P_{cl}^{CPU} is the control place CPUs PC, which can be modelled as:

$$P_{cl}^{CPU} = P_{MME}^{CPU} + P_{SGW}^{CPU} + P_{PGW}^{CPU} \tag{21}$$

The PCs P_{MME}^{CPU}, P_{SGW}^{CPU} and P_{PGW}^{CPU} of the CPUs belong to MME, SGW and PGW, respectively, are equivalent to the sum of their corresponding cores' PCs:

$$P_{MME}^{CPU} = \sum_{k2=1}^{K2} \sum_{c2=1}^{C2} P_{core(k2,c2)}^{MME} \tag{22}$$

$$P_{SGW}^{CPU} = \sum_{k3=1}^{K3} \sum_{c3=1}^{C3} P_{core(k3,c3)}^{SGW} \tag{23}$$

$$P_{PGW}^{CPU} = \sum_{k4=1}^{K4} \sum_{c4=1}^{C4} P_{core(k4,c4)}^{PGW} \tag{24}$$

Where P_{core}^{MME}, P_{core}^{SGW} and P_{core}^{PGW} denote the core PC of MME, SGW and PGW, these are responsible of MME, SGW and PGW functions, respectively. $C1$, $C2$, $C3$ and $C4$ denote the total number of BBUs, MMEs, SGWs and PGWs cores, correspondingly. $K1$, $K2$, $K3$ and $K4$ are the total number of CPUs belong to BBUs, MMEs, SGWs and PGWs, respectively.

4. System Performance loss

The previous description presents how the number of VMs (N) affect server's PC. Herein, the modelling can go further to a single VM level. The degraded performance mentioned in Subsection (1.1) can be modelled regarding the increased server delay and processed RBs within each VM by time. To exceed the complexity of other power models which are either software, heuristic, intrusive and unit utilisation ratio based models, we have correlated the dynamic PC of each VM linearly or convexly with the number of physical resource blocks (RB) or bandwidth processed, which is an easy parameter to obtain [18]. The higher processed RBs of each VM, the more share from the limited server resources is demanded, the more PC. To model the dynamic or linear PC of each VM at each server component, the number of RBs linearly influences the constant base load PCs P_{intRAM}, $P_{intcore}$, P_{intnic} or P_o, these are independent of N or the number of RBs allocated to each VM (n), where $n \in N$. These initial values are increased by the amount ($\sum_n^N e^{\vartheta * RB_n}$), where ϑ is the increment factor due to processing RB_n by n-th VM in any of the server resources. This granularity in the PC is jointly added to the models of Section (2) to produce total PC of the virtualised CN (P_{server}^{vCN}), as follows:

$$P_{server}^{vCN} = [P_{server}^{BBU} + \sum_{s_1}^{S_1} \sum_{n}^{N} e^{\vartheta * RB_n}] + [P_{server}^{cl} + \sum_{s_2}^{S_2} \sum_{n}^{N} e^{\vartheta * RB_n}] \tag{25}$$

The above formulation draws about 40% gain in the PC of each server, which is originated from increasing both VMs and RBs allocation, this fits but not limited to the real time server measurements presented in [5]. Fig. 1 compares the initial PC of a single server and the power cost imported by increasing the number of RBs, VMs and both.

Another performance factor is the time it takes the VM to process these RBs. The execution time of the traditional load in a BBU increases linearly with the number of RBs and the modulation coding scheme ($MCS \in \{9, 16, 25\}$) used to transmit/receive these RBs [18]. A single VM may require 5 times more delay to processed a packet compared to a bare metal LTE BBU; due to increased accessing calls and interrupts between VM-HV and HV-server resources. Modelling this concept requires introducing a factor called MCS index (mcs) to describe the degree of linearity between the RBs and execution time in a bare BBU server (τ_{bare}), as shown in Fig. 2, where $\tau_{bare} = \tau_{init} + (mcs * RB)$, and τ_{init} is the initial BBU delay due to other BBU functions rather than MCS. Subsequently, the HV delay (τ_{HV}) is added to the above description, i.e. $\tau_v = \tau_{bare} + \tau_{HV}$, where τ_v is the execution time of virtualised server when 1 VM is installed, as shown in Fig. 3 for different MCS values. After, the total execution time (τ) of all VMs is expressed as $\tau = \sum_s^S \sum_n^N \tau_v^{s,n}$, where $\tau_v^{s,n}$ denotes the execution time of VM n located in server s.

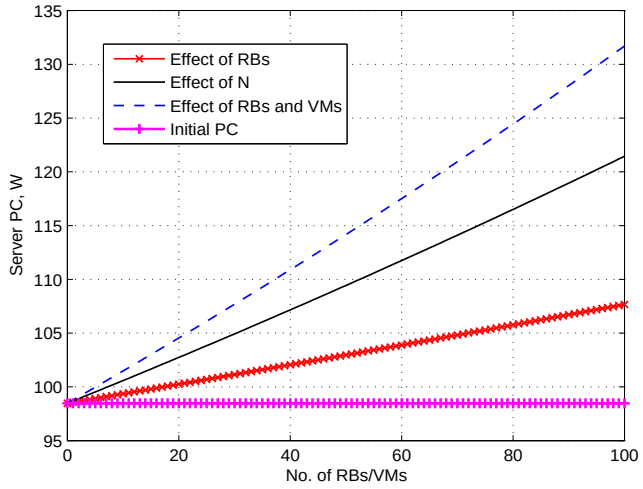

Fig. 1: PC of a BBU server at 100 VMs or RBs, the server's PC increases from initial to about 40% due to both parameters.

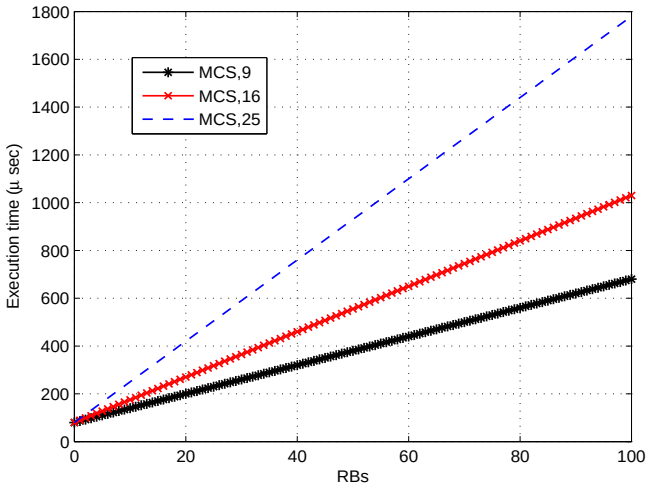

Fig. 2: Execution time of a bare server at different MCS and different number of RBs.

5. Total Power Consumption

Total PC of the CN is conformed to the effects of other losses such as AC-DC, DC-DC and cooling loss in a straightforward manner. This offers an easy but accurate way to calculate their PC without undergoing the computations of each unit. Therefore, AC-DC, DC-DC and cooling PC are linearly scaled with other components' PC and approximated by using loss factors (σ_{DC}, σ_{AC}, σ_{cool}) to represent AC-DC, DC-DC and cooling loss factors, respectively. Successively, the total PC of virtualised C-RAN (P_{vCRAN}), in W, is formulated as follows:

$$P_{vCRAN} = \frac{P_{server}^{vCN}}{(1-\sigma_{DC})(1-\sigma_{MS})(1-\sigma_{cool})} \tag{26}$$

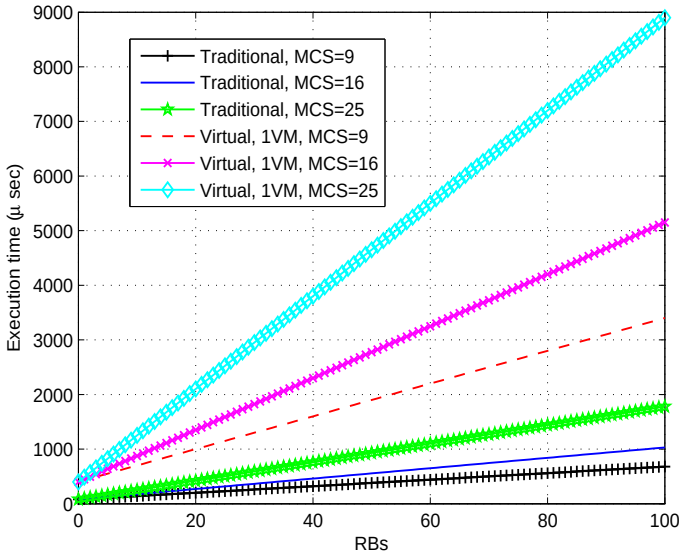

Fig. 3: Comparing the execution time of a bare metal and virtualised server at 1 VM and different MCSs while processing 100 RBs.

6. Results

To correlate the findings of this work with a real time measurement, the resulting parameters were selected from [8], [15],[16], [17], as shown in Table 1. The experimental data related to the PC of each component in the server demonstrates that the initial PC of the CPU is 29.6W, RAM is 4W, NIC is 2W and HDD is 25W, while the rest of PC in the server is resulted from the overhead. While emphasising on the context of showing how the VMs increasingly affect the PC, only $P^{i}_{storage}(t)$ of the storage PC is considered. Furthermore, the second RAM PC model is adapted in the results. To compare the model with particular experimental data without losing the generality of this work, the parameters used in Table 1 have resulted about 40 % PC increment at each virtualised server which fits what is measured in [5]. This increment also synchronised with degraded performance in each server as mentioned in Section (4). Fig. 4 shows a comparison of BBU pool's PC with and without virtualisation for different number of virtualised host servers (i.e., 5, 10, and 20 servers). This compares the cost of processing the maximum LTE bandwidth allowed (100 RBs) by each one of the 100 bare metal BBU servers, and also shows the effect of processing the same amount of RBs by each of the 100 VMs installed in the virtualised servers.

By using the rule of percentage change ($\frac{V1-V2}{|V1|} * 100\%$), it was found that the reductions in the PCs are about 93, 88, and 74 % corresponding to running 5, 10 and 20 virtualised servers compared with the non virtualised servers. On the same basis, Fig. 5 shows a comparison of the entire CN's PC with and without virtualisation. Three control plane servers for MME, SGW and PGW were added to the system while following the previous procedure. This case has resulted a total PC reduction of about 91, 83 and 73 % when respectively running 8, 13, and 23 virtualised server compared to bare metal servers. Continuously, Fig. 6 shows cooling PC comparison of the CN. Cooling PC has been reduced to about 93, 88, and 74 % when compared to the bare metal servers' PC in [10].

It is worth noting that if another type of server with different specifications was used in this

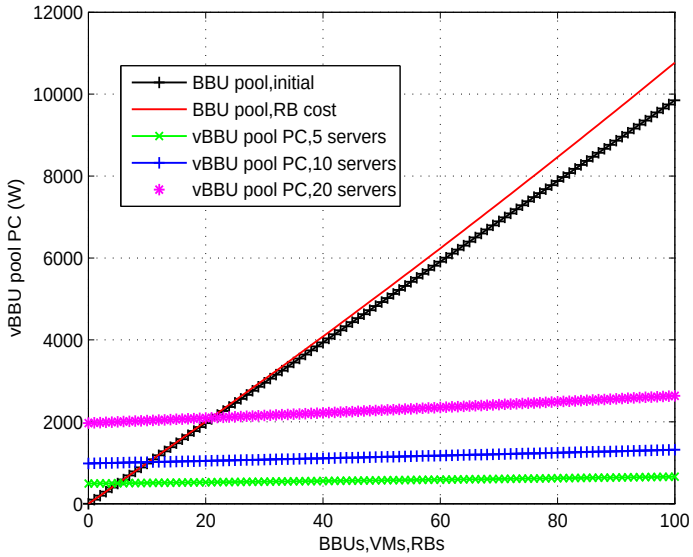

Fig. 4: Comparison of the BBU pool PC with and without virtualisation of 100 BBUs or VMs while processing 100 RBs.

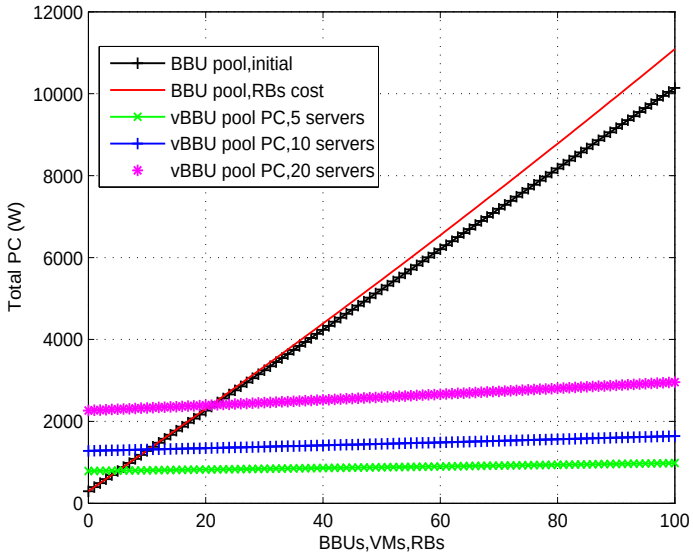

Fig. 5: Comparison the total PC of the CN with and without virtualisation of 100 BBUs, VMs or RBs while processing 100 RBs.

comparison, the eventual results of these comparisons will slightly change. This is because such matter instantly affects the initial and maximum consumption of each component within the virtualised servers, which affects the final outcomes. On the other hand, the more number of bare servers or VMs to be involved in this comparison, the more PC reduction; synchronised with more loss of performance. The reason is that any large number of bare servers will be multiplied by each server's PC. However, in virtualisation method, the main factor (N) is always less than the total

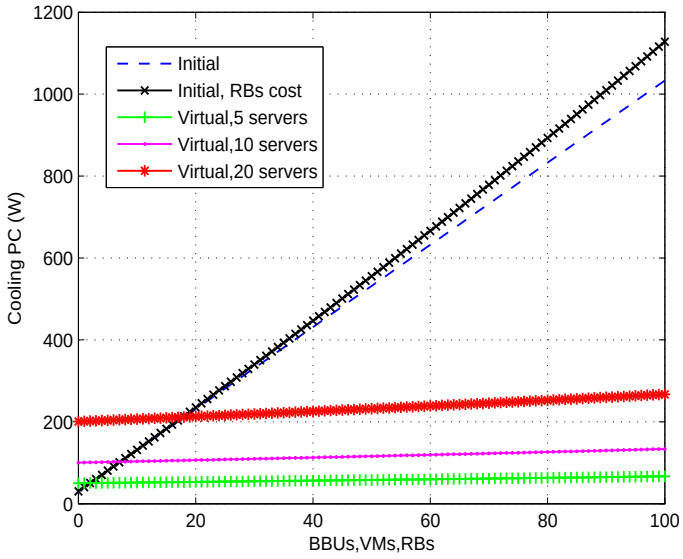

Fig. 6: Comparison of cooling PC of the CN with and without virtualisation of 100 BBUs or VMs while processing 100 RBs.

number of bare servers. This number only influences the number of virtualised servers (S_1 and S_2) while bearing RB cost.

Although the HV's PC is modelled separately than other server's components, its PC considerations are embedded within the model and not jointly added to the total PC value. Once the HV shares host server's units, these chips however must not exceed their maximum power. Accordingly, if the HV PC is separately added to the server's RAM, CPU, etc., this will overtake the real assumptions and the physical realisation regarding the PC of host server's components. This conduct is already considered while modelling each component's PC. Nevertheless, the HV can be considered as one of the VMs that contributes to the PC calculations and inherently included within N. Rather, the effect of the HV up on the execution time a VM consumes to process 100 RBs is shown in Fig. 7. The execution time of a the HV ($\tau_{HV} = 500\mu sec$), which is responsible for 5 times more delay to process the same amount of RBs when the server holds 1 VM, is added to the bare metal server delay $\tau_{bare} = 100\mu sec$. After, the resulting τ_{HV} is multiplied by the number of VMs in the server to obtain the execution time of the virtualised server (τ_v) with 10 VMs, as shown in Fig.7. Furthermore, to extend the delay performance presented in Fig. 7 to the whole system, Fig. 8 compares the total execution time (τ) of three cases regarding the number of bare metal and virtualised servers. These are 5, 10 and 20 virtualised servers, each with 5 VMs, which means there will be 25, 50 and 100 VMs respectively; compared to the same number of bare metal counterparts, all are subjected to ($MCS = 9$). This case increases the delay to about 77, 79 and 80 % accordingly.

In terms of accuracy, the model mainly relies on the manufacturer specifications and design of each component (i.e., components' data sheets). As each equipment holds different operating conditions, such as initial and maximum PC, cooling requirement and efficiency, the outcomes of the model will be accordingly affected. On account of such variation is required to be adjusted through the tuning factors mentioned such as ϑ, β,γ, etc., the power tolerance of each component can be precisely found and added to the total PC of virtual C-RAN. In is worth noticing that if evaluat-

Table 1 MODEL PARAMETERS

Component	Unit	Value
P_{server}^{BBU}	W	29.6
P_{server}^{cl}	W	29.6
P_{intram}^{i}	W	4
$P_{intcore}$	W	3.7
P_{intnic}	W	2
P_o	W	10
τ_{init}	μ sec	80
τ_{HV}	μ sec	500
σ_{DC}	-	0.075
σ_{MS}	-	0.09
σ_{cool}	-	0.1
β	-	0.003
μ	-	0.1
γ	-	0.08
δ	-	-0.001
ξ	-	0.007
ν	-	0.005
ϑ	-	0.001
ε	-	0.009
C	-	4
$mcs, 9$	-	6
$mcs, 16$	-	9.5
$mcs, 25$	-	17

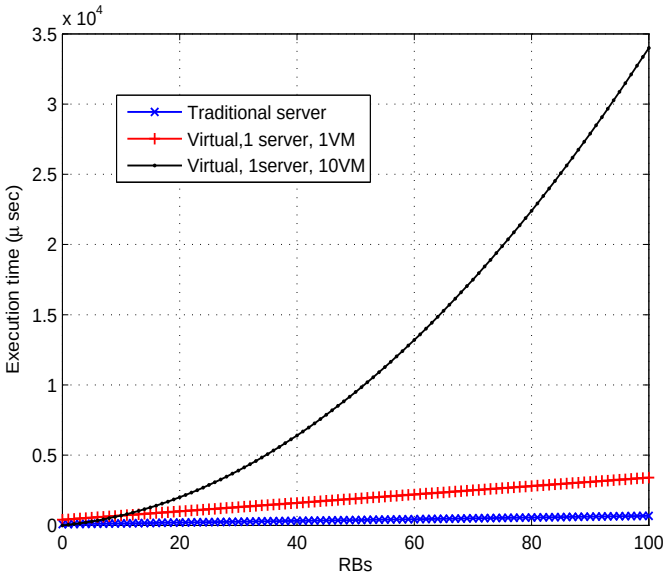

Fig. 7: RBs effect up on the execution time of a single server with and without virtualisation when constituting 1 or 10 VMs.

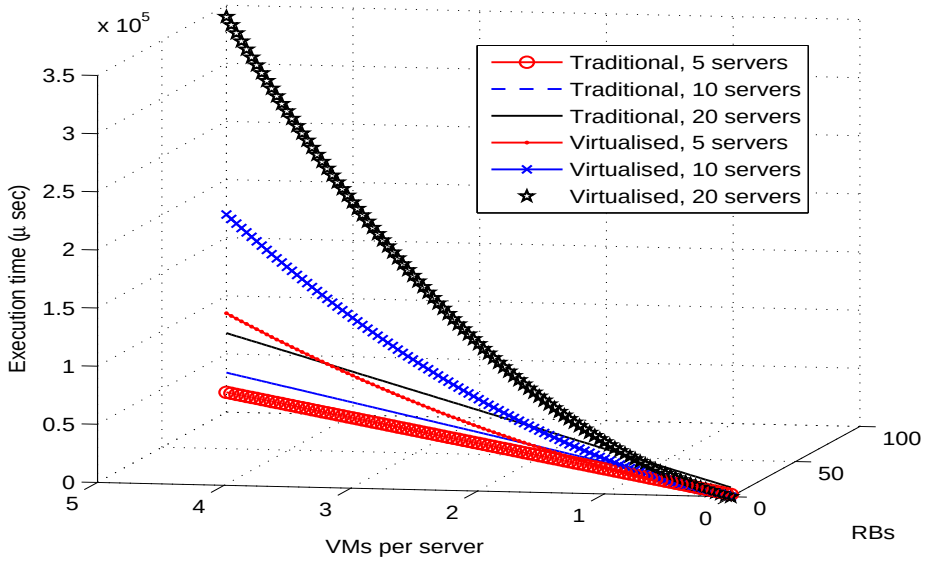

Fig. 8: Comparing VMs and RBs effect up on the execution time of the entire CN with and without virtualisation for different number of servers.

ing systems with different number of VMs and RBs, the tuning parameters can be easily adjusted so as the maximum consumption of the each server's component is reached. Subsequently, total PC (P_{vCRAN}) can be predicted. At present, the model is reliable to judge the virtualised network as it relies upon initials and assumptions come from real data measurements and experimental background as found in [5] and [18].

7. Conclusion

A parameters based PC model has been presented to demonstrate the PC calculations of virtualised C-RAN architecture. By using this model, the power and execution time cost of NFV can be assessed. The model also enables the network providers to distinguish the PC reduction of the entire network while bearing the predicaments of NFV. The cost of increasing the VMs in a virtualised server and processed RBs of each VM is presented by providing a comparison for cooling, execution time performance and total PC with a non virtualised counterparts. The model is adaptable to any varying values related to any server resource, as the factors (γ, β, etc.) and initial PC values used in the model are changeable to describe the increasing/decreasing in PC of each resource found in the server. These latter are subjected to varying manufacturing specification of each electronic chip/device. Intuitively, NFV dramatically reduces network PC. At the same time, it degrades the execution time efficiency of the virtualised servers while preforming network's functions. To justify such issue, there have been recently several works proposed to optimise the problems of the HVs' scheduling procedure. These problems are related to synchronization, real-time constraints, security, VMs placement and performance enhancement. However, these researches are proposed to unleash and extend the HV capability to an optimum and enhance the lack of performance in a vitrualised server/network in real time services, such as in [19], [20] and [21]. The investigations also promise new futuristic techniques, protocols, algorithms and designs to be innovated. However, the benefits and characteristics gained by reducing the operational cost and total PC of the network compensate such performance loss in case of advanced techniques are used to mitigate the HV's constraints. Therefore, this work advocates the use of virtualisation in the coming generations as a way to greatly reduce the PC.

8. References

[1] Peng, M., Li, Y., Zhao, Z., and Wang, C.,: "System architecture and key technologies for 5G heterogeneous cloud radio access networks,", IEEE Network, 2015, **29**, (2), pp. 6–14.

[2] Katsalis, K., et al., "5G Architectural Design Patterns,", IEEE International Conference on Communications Workshops (ICC), 2016, pp. 32-37.

[3] Demestichas, P., et al., "5G on the Horizon: Key Challenges for the Radio-Access Network,", IEEE Vehicular Technology Magazine, 2013, **8**, (3), pp. 47–53.

[4] Costa-Perez, et al.: "Radio access network virtualization for future mobile carrier networks,", IEEE Communications Magazine, 2013, **51**, (7), pp. 27–35.

[5] Shea, R., Wang, H., Liu, J.: 'Power consumption of virtual machines with network transactions: Measurement and improvements', IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, 2014, pp. 1051–1059.

[6] Alhumaima, R. S., Al-Raweshidy, H. S.: 'Evaluating the energy efficiency of software defined-based cloud radio access networks', Commun. IET, 2016, **10**, (8), pp. 987–994.

[7] Ricardo, L.: 'Evaluating the performance and power consumption of systems with virtual machines', Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International

Conference, 2011, pp. 778–783.

[8] Auer, G., Giannini, V., Desset, C., et al.: 'How much energy is needed to run a wireless network?', Wirel. Commun. IEEE, 2011, **18**, (5), pp. 40–49.

[9] Holtkamp, H., Auer, G., Giannini, V., Haas, H.: A Parameterized Base Station Power Model, Commun. Lett. IEEE, 2013, **11**, (17), pp. 2033-2035.

[10] Alhumaima, R. S., Khan, M., Al-Raweshidy, H. S.: 'Component and parameterised power model for cloud radio access network', Commun. IET, 2016, **10**, (7), pp. 745–752.

[11] Li, Y., Chen, M.: 'Software-Defined Network Function Virtualization: A Survey', Access IEEE, 2015, **3**, pp. 2542–2553.

[12] Huang, Q., Gao, F., Wang, R., Qi, Z.: 'Power Consumption of Virtual Machine Live Migration in Clouds', Communications and Mobile Computing (CMC), 2011 Third International Conference on, (2011), pp. 122–125.

[13] Marcu, M., Tudor, D.: 'Power consumption measurements of virtual machines', Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium, pp. 445–449.

[14] Dayarathna, M., Wen, Y. and Fan, R.: "Data Center Energy Consumption Modeling: A Survey", IEEE Communications Surveys and Tutorials, 2016, **18**, (1), pp. 732–794.

[15] Ye, L., Gniady, C., Hartman, J. H.: 'Energy-efficient memory management in virtual machine environments', Green Computing Conference and Workshops (IGCC), (2011), pp. 1–8.

[16] Maurizio, P., Christian, E.: 'Network consolidation for virtualized servers', US Patent, **8**, 2011.

[17] Lauri, M., Brad, E.: 'The problem of power consumption in servers', Intel Corporation. Dr. Dobb's, 2009.

[18] Sourjya, B., et al.: "CloudIQ: a framework for processing base stations in a data center.", 18th annual international conference on Mobile computing and networking (ACM), (2012), pp. 125–136.

[19] Wu, S., Zhou, L., Sun, H., Jin, H., Shi, X.: 'Poris: A Scheduler for Parallel Soft Real-Time Applications in Virtualized Environments', Transactions on Parallel and Distributed Systems IEEE, 2016, **27**, (3), pp. 841–854.

[20] Kim, C., Park, K. H.: 'Credit-Based Runtime Placement of Virtual Machines on a Single NUMA System for QoS of Data Access Performance', Transactions on Computers IEEE, 2015, **64**, (6), pp. 1633–1646.

[21] Zhang, M., Yang, Q., Wu, C. and Jiang, M., "Hierarchical virtual network mapping algorithm for large-scale network virtualisation,", Commun. IET, 2012, **6**, (13), pp. 1969–1978.