Energy Proportionality in Near-Threshold Computing Servers and Cloud Data Centers: Consolidating or Not?

Ali Pahlevan*, Yasir Mahmood Qureshi*, Marina Zapater*, Andrea Bartolini[†‡], Davide Rossi[‡],
Luca Benini[†‡], David Atienza*

*Embedded Systems Laboratory (ESL), Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland
[†]Swiss Federal Institute of Technology Zurich (ETHZ), Switzerland
[‡]University of Bologna (Unibo), Italy
*{ali.pahlevan, yasir.qureshi, marina.zapater, david.atienza}@epfl.ch, [†]{barandre, lbenini@}@iis.ee.ethz.ch, [‡]{davide.rossi}@unibo.it

Abstract—Cloud Computing aims to efficiently tackle the increasing demand of computing resources, and its popularity has led to a dramatic increase in the number of computing servers and data centers worldwide. However, as effect of post-Dennard scaling, computing servers have become power-limited, and new system-level approaches must be used to improve their energy efficiency. This paper first presents an accurate power modelling characterization for a new server architecture based on the FD-SOI process technology for near-threshold computing (NTC). Then, we explore the existing energy vs. performance trade-offs when virtualized applications with different CPU utilization and memory footprint characteristics are executed. Finally, based on this analysis, we propose a novel dynamic virtual machine (VM) allocation method that exploits the knowledge of VMs characteristics together with our accurate server power model for next-generation NTC-based data centers, while guaranteeing quality of service (QoS) requirements. Our results demonstrate the inefficiency of current workload consolidation techniques for new NTC-based data center designs, and how our proposed method provides up to 45% energy savings when compared to state-of-the-art consolidation-based approaches.

I. Introduction

Cloud computing has recently been brought into focus in both academia and industry due to the increase of applications and services. Consequently, there has been a rapid growth in the number of data centers in the world, leading to unsustained energy consumption, estimated to be at 1.3% of the global energy usage, and growing at a yearly rate of 20% [1].

To maximize energy efficiency (i.e., performance per watt), customized server architectures increase throughput by identifying and eliminating the bottlenecks of conventional server processors. However, as an effect of post-Dennard scaling [2], energy reduction in deep sub-micron technologies has lagged behind, resulting in power-limited servers.

A promising approach to overcome the power bottlenecks is near-threshold computing (NTC). NTC takes advantage of the quadratic dependency between supply voltage and dynamic power consumption, by lowering the operating voltage to a value slightly higher than the transistor threshold, increasing energy efficiency at the expense of reduced performance. However, for current cloud applications, NTC allows to optimize the trade-off between performance and power, emerging as a promising approach to overcome the power-wall [3].

From a technology viewpoint, the ultra-thin body and buried oxide (UTBB) FD-SOI technology has demonstrated its suitability for NTC. In contrast to traditional bulk technology, FD-SOI features a significantly increased voltage range and even higher performance for the same energy thanks to the better behavior of transistors at low voltage [4]. The 28nm FD-SOI technology process is currently employed for mass production by Samsung and ST Microelectronics; the 20nm technology is being produced by GlobalFoundries while the 12nm node is on the strategic roadmap [5]. With respect to FinFET technology, FD-SOI provides a cost-sensitive solution for low-power (both active and leakage) systems without increasing die cost [6], making it a suitable solution for next-generation NTC servers.

To the best of our knowledge, the new trade-offs brought by the FD-SOI technology and NTC servers, and the analysis of its impact on data center level energy-aware policies, remains an open challenge. Virtual machine (VM) consolidation [7] has represented for years the most widely used technique to minimize energy consumption. However, the emergence of energy-proportional NTC servers, with drastically reduced static power, together with the advent of applications able to work at reduced frequencies, changes the underlying assumptions that made consolidation best for energy efficiency.

In this paper we propose accurate power models for NTC servers, and evaluate their impact of VM consolidation. We demonstrate the stagnation of consolidation-based and server turn-off techniques for NTC-based data centers, and propose a policy that provides 45% energy savings when compared to state-of-the-art consolidation approaches.

In particular, the contributions of our work are as follows:

- We present an accurate power model for the UTBB FD-SOI process technology in NTC servers, together with a power and performance validation against real servers (Intel x86 and ARM64), and propose a performance-improved architecture for NTC servers.
- We show how the energy-proportionality of NTC servers, enabled by the FD-SOI technology, results in a paradigm shift in which traditional VM consolidation strategies no longer yield the optimal results in energy consumption.
- We propose the Energy Proportionality-Aware dynamiC allocaTion (EPACT) method, a novel data center workload allocation policy for NTC servers, which also selects

the best dynamic voltage and frequency scaling (DVFS) setup. Our approach increases the energy proportionality of NTC-based data centers, outperforming latest consolidation techniques, while guaranteeing quality of service (QoS) requirements.

- We assess the performance and efficiency of virtualized workloads on three architectures: (i) x86, (ii) ARM-based Cavium ThunderX, and (iii) our proposed NTC server, which modifies and improves the efficiency of the ThunderX architecture.

II. RELATED WORK

A. Technology and Architecture

Recent work in the area of energy-efficient server design focuses on presently-shipping enterprise servers, with traditional x86 architectures [8]. These servers had traditionally been designed to meet performance goals, without energy efficiency as a design constraint. Only recently, with the stagnation of Dennard Scaling [2], and the resulting power-limited servers, NTC turned into a key technology to improve energy efficiency. Previous work on near-threshold manycores mainly focused on single voltage domain and multiple frequency domain architectures [9]. However, other recent works on processors in FD-SOI demonstrated the near-threshold capabilities of the technology, capable to run a dual-core CortexA9 processor at 1 GHz at the supply voltage of 0.6V [4]. The work presented in [3] was the first one proposing the usage of NTC servers in UTBB FD-SOI technology. Nonetheless, the power model proposed in that work did not include a detailed characterization of the uncore components. Moreover, the target Cavium ThunderX servers [10] were neither based on FD-SOI technology nor validated for virtualized applications.

B. Energy-aware VM Allocation

Research in the area of energy efficiency in cloud computing usually focuses on consolidation-based VM allocation techniques to decrease power while meeting a certain QoS [11]. When deciding the allocation of VMs to physical servers, several works only check that the total size of VMs' load does not exceed the maximum server's capacity [7], [12], or their peak, off-peak, and average utilization of VMs [13], [14]. However, the dynamic nature of cloud workloads results in the CPU-load correlation across VMs (i.e, the similarity of CPU utilization traces and the coincidence of their peaks) [15]. In this context, a few studies [16], [17] consider CPU-load correlation to achieve further energy savings. In particular, Verma *et al.* [16] define VMs' CPU utilization in a time series as a binary sequence. However, this quantization alters the original behavior and is only applicable when VM envelops are stationary. Kim *et al.* [17] present a CPU-load correlation-aware solution to separate CPU-load correlated VMs. They also exploit DVFS to achieve further energy savings. Ruan *et al.* [18] propose a dynamic migration-based VM allocation method to achieve the optimal balance between server utilization and energy consumption., while Garg *et al.* [19] tackle the allocation problem for different types of applications to maximize the resource utilization and profit. Nevertheless, having considered the traditional x86 server architectures,

these approaches assume a linear power-frequency relation for a given workload, and large static server power. However, this is not compatible with novel server architectures.

To the best of our knowledge, the exploration of the new trade-offs and impact on energy-aware VM allocation, brought by new server architectures (in particular NTC servers), remains today an open issue. In this paper, we propose a novel dynamic VM allocation method that exploits the knowledge of VMs characteristics and uses our proposed power model to increase the energy proportionality of next-generation NTC-based data centers, while guaranteeing the QoS. Our results demonstrate the inefficiency of the latest workload consolidation techniques for new NTC-based data center designs.

III. OVERVIEW OF THE PROPOSED SYSTEM

A. Server and Data Center Architecture

As a starting point for our server architecture, we chose the Cavium ThunderX platform [10]. However, for our target applications, the Cavium performance was slower (from 1.5x to 1.35x) than the x86 platform with similar characteristics, and unable to meet QoS constraints, as shown in Table I in Section VI-A. This was due to an inappropriate memory subsystem design for the target applications considered and the choice of in-order cores. Hence, we modified the original architecture and used ARMv8 Cortex-A57 Out-of-Order (OoO) cores, instead of the in-order Cortex A53 processor. We model a 16-core CPU (instead of 48 in ThunderX) to achieve a lower simulation turnaround time, as we experimentally observed that our model linearly scaled up for larger servers. The memory subsystem was also updated, by including L1 instruction cache (I-cache) and data cache (D-cache) of 64KB and 32KB, respectively. A Last-Level-Cache (LLC) of 16MB is modeled. A total memory size of 16GB is considered using a DDR4 memory model with memory controller [20]. DDR4 is clocked at 2400MHz with a peak bandwidth of $19.2GB/s$. Without loss of generality, we model for this exploration a data center with 600 NTC servers, each NTC server with its dedicated power supply, fans and disks.

B. Application Description

Our applications consist of VMs, virtualized via Linux LXC containers, and running synthetically generated workloads that resemble batches of real banking applications, as reported by our industry partners. For realistic CPU and memory usage, we use one week of traces of Google Cluster [21], which provides the CPU and memory utilization for over 600 VMs, reported every 5 minutes (memory utilization is varying in the range of 2% to 32%). Therefore, for profiling purposes, we split the workloads in three categories, according to the per-VM memory utilization: i) low-mem for average memory usage of 70MB (7%), ii) mid-mem for 255MB (25%) and iii) high-mem for 435MB (43%). Moreover, in order to run the experiments in worst-case scenarios, we tune the workloads to maximum CPU utilization.

C. QoS Degradation Constraint for VMs

Because banking applications are virtualized batch jobs, their QoS constraints are defined in terms of the maximum

allowable degradation (i.e., increase in their execution time), which in our case is defined as 2x [8], w.r.t. a baseline execution in a 16-core Intel Xeon X5650 running at 2.6GHz, with 12MB LLC and 128GB of RAM clocked at 1333MHz, in which we run one LXC container (VM) per core.

IV. SERVER AND DATA CENTER POWER MODELS

The overall NTC server power model has been extracted by combining direct measurement on a commercial ARM-v8 based server [22] with power measurements of real prototypes implemented in 28nm FD-SOI technology and operating in near-threshold [4], [23], allowing for a very accurate system power estimation for all the operating conditions investigated in this work. We consider four main contributors to the overall power consumption of the server: 1) the core region composed of the A57 cores logic and the L1/L2 caches, 2) the LLC, 3) the memory controller, peripherals, IO subsystem and motherboard, and 4) the DRAM banks.

1) Cores: Similarly to [3] we combine the 28nm FD-SOI power and performance model of a recent Cortex A9 implementation of STM in 28nm bulk and FD-SOI, considering the differences in pipeline length ratio and critical path between Cortex A57 and Cortex A9. These parameters are extracted by comparing the different voltage to frequency ratio (extracted via the CPUFreq Linux driver) present in the Samsung Exynos processor family. The Cortex A57 is 1.17x faster (higher-frequency) than the Cortex A9. We combine this information with the active and static energy per clock cycle at the different DVFS levels from the Samsung Exynos 5433 processors to scale its energy figures to the STM 28nm FD-SOI technology by using the trends reported in [4]. These numbers account also for the L1 and L2 cache power consumption. When in wait-for-memory (WFM) state the core region consumes 24% less power than when active. This number has been measured empirically on an Intel Xeon v3 processor. Then, we extended the performance and power model to the NTC region fitting a template extracted from measurements of a 28nm UTBB FD-SOI near-threshold parallel processor [23].

2) Last-Level-Cache (LLC): The LLC power model was extracted by measuring the leakage power for a 256KB SRAM block in 28nm UTBB FD-SOI and read and write energy [pJ/Access] for 128-bit wide accesses. All these values have been obtained for different voltage levels.

3) Memory Controller, Peripherals, IO and Motherboard: We empirically measure the memory controller, peripherals and IO subsystem power consumption overhead of an Intel Xeon v3 CPU. This power consumption is split in two parts: (i) a constant component which accounts for the static and fix dynamic power cost needed to keep these subsystems on, and (ii) a component proportional to the operating condition. The constant part causes a 11.84W overhead in all operating points, while the proportional one ranges from 9 to 1.6W in the operational range. Finally, we assume the same motherboard power consumption than in the Cavium ThunderX server, which is of 15W for a low fan speed, and with 1 SSD disk.

4) DRAM: The DRAM power has been modeled with direct measurement on a real server platform based on Intel Xeon
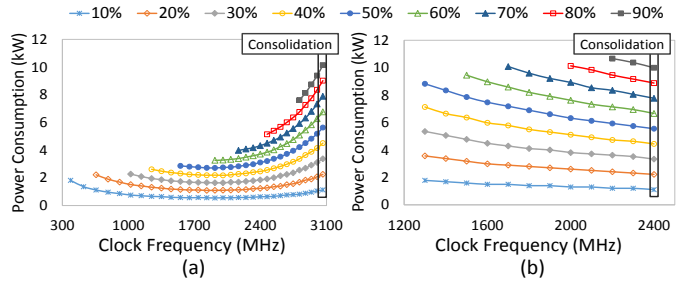

Fig. 1: Power consumption under different data center utilization for CPU-bounded tasks (no dynamic memory power) for (a) NTC-based and (b) non-NTC-based data center.

v3 architecture. During a large variety of workloads we have measured the total DRAM read and write accesses in windows of 1 second and measured the power of the DRAM banks. Afterwards, we interpolated the empirical measurement with a linear power model. The final model contains the empirical measurement of an idle power value of 15.5 [mW/GB] per GB of DRAM, which increases to 155 [mW/GB] when the banks are activated. On top of this static power we reported an energy consumption of 800 [pJ/Byte] per byte read.

5) Overall Data Center power: All these power consumption values have been inserted in the GEM5 simulator to estimate the power consumption of each server node under real workload. We model total data center power consumption (P_{DC}) as the sum of power consumed by servers (P_s).

V. PROPOSED OPTIMIZATION METHOD

A. Data Center Scenario and Motivation

Figure 1(a) shows the worst-case data center power consumption in an NTC-based data center when servers run at different frequencies for various data center CPU utilization rates, defined as the ratio of required CPU resources in MHz to the total CPU resources (i.e., the number of servers multiplied by the maximum CPU resources of one server), when running a CPU-bounded workload (i.e., dynamic memory power is close to zero). In this setup, we consider 80 servers with a maximum frequency (F_{max}) of 3.1GHz. As CPU utilization rate increases, we need to either turn on more servers, or set higher frequencies to the turned-on servers. A traditional consolidation approach minimizes the amount of active servers and runs them at the highest frequency possible. However, in NTC-based data centers, we observe that the optimal frequency of servers (F_{opt}^{NTC}) is around 1.9GHz, instead of 3.1GHz, due to the non-linear behavior of CPU power with frequency. For a utilization rate higher than 50%, the optimal frequency is the minimum possible that meets the workload demand. On the contrary, Fig. 1(b) shows the power consumption of a non-NTC server data center (equipped with 6-core Intel E5-2620 servers), representing the efficiency of consolidation.

On the other hand, in our proposed model, memory power consumption is a linear function of the number of memory accesses per second. Thus, from the memory power perspective, to minimize energy consumption we should consolidate as many VMs as memory allows, and keep the number of active servers to a minimum. Hence, in NTC-based data centers, CPU

Algorithm 1 The Proposed 1D VM Allocation Algorithm

Input: $\tilde{U}_{cpu}$, $\tilde{U}_{mem}$, F_{opt}, and F_{max}
Output: Allocating VMs to servers
1: $ID_s \leftarrow 1$
2: **while** All VMs not allocated **do**
3:     **if** Server ID_s is empty **then**
4:         $ID_{VM} \leftarrow$ First unallocated VM
5:         $Patt_{ID_s,cpu} \leftarrow Patt_{ID_s,cpu} + \tilde{U}_{cpu}^{ID_{VM}}$
6:         $Patt_{ID_s,mem} \leftarrow Patt_{ID_s,mem} + \tilde{U}_{mem}^{ID_{VM}}$
7:     **else if** Server ID_s is not empty **then**
8:         $Patt_{ID_s,cpu}^{Com} \leftarrow \max(Patt_{ID_s,cpu}) - Patt_{ID_s,cpu}$
9:         **for** $i = 1$: Number of unallocated VMs **do**
10:             $\phi_i \leftarrow$ PearsonCorrelation($Patt_{ID_s,cpu}^{Com},\tilde{U}_{cpu}^i$)
11:         **end for**
12:         Find VM (ID_{VM}) with maximum ϕ & $\max(Patt_{ID_s,cpu} + \tilde{U}_{cpu}^{ID_{VM}}).F_{max} \leq F_{opt}$
13:         **if** $ID_{VM} == Null$ **then**
14:             $ID_s \leftarrow ID_s + 1$
15:         **else**
16:             Allocate VM ID_{VM} to server ID_s
17:         **end if**
18:     **end if**
19: **end while**

Algorithm 2 The Proposed 2D VM Allocation Algorithm

Input: $\tilde{U}_{cpu}$, $\tilde{U}_{mem}$, Cap_{cpu}, and Cap_{mem}
Output: Allocating VMs to servers
1: **for** $i = 1 : N_{VM}$ **do**
2:     **for** $j = 1 : N_s$ **do**
3:         **if** $\max_n(\tilde{U}_{cpu}^{i,n} + S_{cpu}^{j,n}) \leq Cap_{cpu}$ & $\max_n(\tilde{U}_{mem}^{i,n} + S_{mem}^{j,n}) \leq Cap_{mem}$ **then**
4:             \\ CPU
5:             $Patt_{j,cpu}^{Com} \leftarrow \max(S_{cpu}^j) - S_{cpu}^j$
6:             $\phi_{cpu}^{j,i} \leftarrow$ PearsonCorrelation($Patt_{j,cpu}^{Com},\tilde{U}_{cpu}^i$)
7:             $S_{rem,cpu}^j \leftarrow Cap_{cpu} - S_{cpu}^j$
8:             $Dist_{cpu}^{j,i} \leftarrow \|\tilde{U}_{cpu}^i - S_{rem,cpu}^j\|_2$
9:             \\ Memory
10:             $Patt_{j,mem}^{Com} \leftarrow \max(S_{mem}^j) - S_{mem}^j$
11:             $\phi_{mem}^{j,i} \leftarrow$ PearsonCorrelation($Patt_{j,mem}^{Com},\tilde{U}_{mem}^i$)
12:             $S_{rem,mem}^j \leftarrow Cap_{mem} - S_{mem}^j$
13:             $Dist_{mem}^{j,i} \leftarrow \|\tilde{U}_{mem}^i - S_{rem,mem}^j\|_2$
14:             $\mathcal{M}_j^i \leftarrow$ Compute efficiency using Eq. 2
15:         **end if**
16:     **end for**
17:     $ID_s \leftarrow$ Find server with max $\mathcal{M}^i$
18:     Allocate VM i to server ID_s and update server's resources
19: **end for**

---

and memory bounded workloads exhibit opposite behaviour in terms of efficiency. Therefore, neither VM consolidation nor load balancing are the best options, as the optimal server frequency and workload allocation strategy dynamically change depending on data center workloads.

B. EPACT: Proposed VM Allocation Method

Given the previous analysis, we propose the Energy Proportionality-Aware dynamiC allocaTion (EPACT) method to allocate the total number of VMs available in the data center (N_{VM}) to servers every time slot T, while trying to make servers work at the most energy-efficient frequency (F_{opt}^T) in each sampled value $1..n$ (one sample every 5 minutes) during time slot T (considered as 1 hour).

Our method requires predicting, at the beginning of T, the per-VM CPU and memory utilization patterns ($\tilde{U}_{cpu}$ and $\tilde{U}_{mem}$). Given the daily periodicity observed in the VMs of Google Cluster traces, we use the autoregressive integrated moving average (ARIMA) prediction model [24]. ARIMA considers the CPU and memory utilization from the previous week and forecasts the next-day traces per VM.

Given these predictions, we first determine the optimal number of turned-on servers from the CPU and memory perspective, independently:

$$\hat{N}_{server}^{cpu} = \frac{\max_n(\sum_{k=1}^{N_{VM}} \tilde{U}_{cpu}^{k,n}) \cdot F_{max}}{F_{opt}^{NTC} \cdot 100}, \; \hat{N}_{server}^{mem} = \frac{\max_n(\sum_{k=1}^{N_{VM}} \tilde{U}_{mem}^{k,n})}{100} \quad (1)$$

From the CPU viewpoint we choose the number of servers that allows to set a frequency as close to $F_{opt}^{NTC} = 1.9GHz$ as possible, and from the memory standpoint we try to consolidate as many VMs as possible until we hit the maximum server memory capacity (i.e., memory cap). The definition of $\hat{N}_{server}^{cpu}$ and $\hat{N}_{server}^{mem}$ results in two cases.

1) If $\hat{N}_{server}^{cpu} > \hat{N}_{server}^{mem}$, we exhaustively explore all the number of turned-on servers between these two values, until we find the F_{opt}^T that exhibits the lowest data center power consumption. Then, as described in Alg. 1, we find the best VMs

fit into servers by using the First-Fit-Decreasing algorithm, only taking into account the CPU utilization, as they drive QoS. Thus, we select one server (ID_s, line 1). If the server is empty, we select the first unallocated VM from the pool of VMs, we allocate it to the corresponding server, and update the server load pattern ($Patt_{ID_s,cpu}$ and $Patt_{ID_s,mem}$, lines 4-6). Otherwise, we first compute the complementary utilization pattern of server ($Patt_{s,cpu}^{Com}$) with respect to its current maximum load (line 8). Then, we select one VM from the pool of VMs which has the maximum similarity (ϕ, defined as the Pearson Correlation) to the pattern such that the maximum aggregated load of server ($\max([Patt_{ID_s,cpu} + \tilde{U}_{cpu}^{ID_{VM}}]/100).F_{max}$) is less than F_{opt}^T (simply named F_{opt}). Otherwise, we turn on another server (lines 9-17).

2) If $\hat{N}_{server}^{cpu} \leq \hat{N}_{server}^{mem}$, memory dominates and the optimal frequency is defined based on $F_{opt} = \frac{\max_n(\sum_{k=1}^{N_{VM}} \tilde{U}_{cpu}^{k,n}) \cdot F_{max}}{\hat{N}_{server}^{mem} \cdot 100}$. In this case, our allocation phase needs to take into account both the CPU utilization and memory footprint patterns to find the best VMs fit into the servers based on CPU cap (i.e, $Cap_{cpu} = (F_{opt} \cdot 100)/F_{max}$) and memory cap (i.e., $Cap_{mem} = 100\%$).

Having chosen the number of servers, we find the best server for each VM, maximizing the following merit function:

$$\mathcal{M}_j^i = \omega_{cpu} \cdot \frac{\phi_{cpu}^{j,i}}{Dist_{cpu}^{j,i}} + \omega_{mem} \cdot \frac{\phi_{mem}^{j,i}}{Dist_{mem}^{j,i}}$$
$$\omega_{cpu} = \frac{Cap_{cpu}}{Cap_{cpu} + Cap_{mem}} \; \& \; \omega_{mem} = \frac{Cap_{mem}}{Cap_{cpu} + Cap_{mem}} \quad (2)$$

where $\phi_{cpu}^{j,i}$ and $\phi_{mem}^{j,i}$ exhibit the similarity of ith VM's CPU utilization and memory footprint patterns with complementary CPU utilization and memory patterns of jth server, respectively. However, as the Pearson Correlation cannot reflect the closeness of VM's CPU and memory patterns to the server CPU and memory cap, respectively, we incorporate the euclidean distance ($Dist_{cpu}^{j,i}$ and $Dist_{mem}^{j,i}$) into the metric. As a

result, Eq. 2 demonstrates that $\mathcal{M}_j^i$ is high when i^{th} VM has both the same shape and lower distance to j^{th} server's caps. ω_{cpu} and ω_{mem} are weighting factors that need to be set with respect to the determined CPU and memory cap for filling up the server resources with the same importance.

As described in Alg. 2, we first select one VM (i^{th} VM) and target to find the best server among all (N_s) for it. For each candidate server (j^{th} server), we check whether it has enough resources for hosting the VM at each sample in time slot T. If the server has sufficient remaining CPU and memory capacity ($S_{rem,cpu}^j$ and $S_{rem,mem}^j$), we compute $\mathcal{M}_j^i$ for the server. Finally, we allocate the VM to the server which has the maximum $\mathcal{M}^i$, and update the target server's resources (lines 17 and 18).

After allocation, for both cases, based on the real VMs CPU utilization, we online set the best frequency level for each server per sample to guarantee QoS.

VI. EXPERIMENTAL RESULTS

A. Simulation Framework Validation

We use the GEM5 cycle-accurate simulator [25] to simulate the server architecture described in Section III-A. In order to understand the effect of DVFS on performance, we compute the QoS degradation taking as a baseline the execution time on the x86 server discussed in Section III-C. Then, we simulate the virtualized applications in GEM5 for different frequency levels ranging from 2.5GHz down to 100MHz.

To validate the correctness of the results provided by the GEM5 simulator, we run the applications on two real hardware platforms based on x86 and ARM. We compared the execution times of Cavium ThunderX with the ones obtained via GEM5 while matching the exact same architectural configuration. The error obtained was below 10%, showing that GEM5 is able to accurately simulate our workloads. The execution time for each workload, on all three platforms are shown in Table I. The QoS limit is a 2x degradation of the execution time on x86 based platform, as already discussed. The Cavium server exhibits the worst execution time. After the modifications undertaken, our proposed NTC server architecture outperforms Cavium by a factor of 1.25x to 1.76x. These results are due to our improved memory sub-system and the incorporation of the OoO processor in our proposed architecture.

B. Server-level Results

1) Quality of Service: QoS requirements of virtualized applications make it unclear whether this technology is suitable for server processors. To check for QoS requirements being met for VM workloads on NTC server, normalized execution time to QoS limit is shown in Fig. 2. It can be seen that high-mem and mid-mem workloads meet QoS requirement till a minimum frequency of 1.8GHz, whereas low-mem can scale down to 1.2GHz. In conclusion, we are able to reduce

TABLE I: NTC server and Cavium ThunderX QoS analysis

Application	Intel x86 @2.66 GHz	2x Degrad. Intel (QoS limit)	Cavium @2GHz	NTC Server @2GHz
low-mem	0.437	0.873	0.733	0.582
mid-mem	1.564	3.127	5.035	2.926
high-mem	3.455	6.909	11.943	6.765

the frequency of the cores while meeting the 2x degradation constraint for virtualized applications.

2) Energy Efficiency: Fig. 3 shows the benefits of reducing DVFS on server energy efficiency (i.e., the total number of UIPS at the chip level, divided by the total power consumption of the server). The optimal efficiency point is around 1.2GHz for high-mem, and around 1.5GHz for low-mem and mid-mem. The energy efficiency decreases with increasing memory utilization, firstly, because of higher active memory power, and secondly, because more memory accesses increase the amount of stalls and the WFM cycles.

3) Trade-offs Discussion: As shown by [3], workloads can tolerate low frequencies if only core power is considered, thus enabling NTC operation to reduce core power consumption. However, not all server components scale with the core voltage, shifting the most energy-efficient point to a higher frequency. Our results showed that frequency can be reduced to 1.2GHz for high-mem and 1.5GHz for low-mem and mid-mem. But, to guarantee the QoS requirements, the frequency level should be scaled up to 1.8GHz for mid-mem and high-mem; while for low-mem (CPU-bounded tasks) the optimal frequency (i.e., 1.5GHz) still meets the QoS limit.

C. Data Center-level Results

At the data center level, we compare our EPACT policy against two other energy-aware methods:

- COAT: COnsolidation-Aware allocaTion [17].
- COAT-OPT: COAT with an OPTimal fixed cap (optimal server frequency) when the worst-case data center power consumption is minimum.

and we evaluate our approach in terms of service level agreement (SLA) violations and overall energy consumption.

1) SLA Violation: Figure 4 shows violation, defined as the number of overutilized servers (i.e., the aggregated CPU or memory utilization among co-located VMs is beyond the CPU and memory cap), during each time slot for a time horizon of one week. Violations can only occur due to miss-prediction on the VM usage, especially during abrupt workload changes. EPACT provides a drastic reduction of the violations compared to COAT and COAT-OPT. This is because, in EPACT, we do not fill up the servers to their maximum capacity, and we have some slack to increase frequency and compensate for violations. On the contrary, COAT and COAT-OPT have less control on violations during peak loads using a fixed cap.

2) Energy Consumption Analysis: Figure 5 shows the number of active servers per time slot for a time horizon of one week. COAT, being consolidation-based, reduces the number active servers by 37% on average compared to EPACT. Despite this fact, EPACT achieves 45 and 10% energy savings in the best and worst case compared to COAT and COAT-OPT, respectively (Fig. 6). This is because the optimal frequency is dinamycally found w.r.t the time-varying data center CPU utilization and memory footprint, thus showing the inefficiency of consolidation-based techniques for NTC-based data centers.

3) Different Amount of Static Power Analysis: Figure 7 represents the effectiveness of our algorithm compared to consolidation-based technique with maximum cap (COAT)
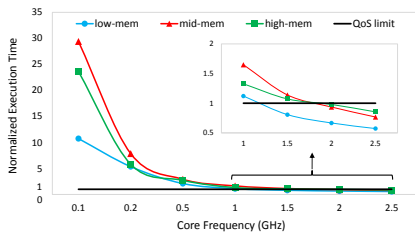
Fig. 2: Execution time normalized to QoS limit for different workloads.
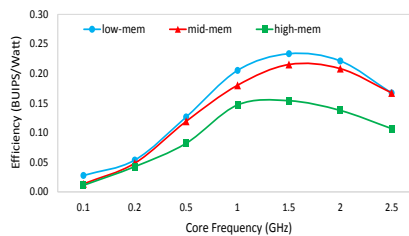

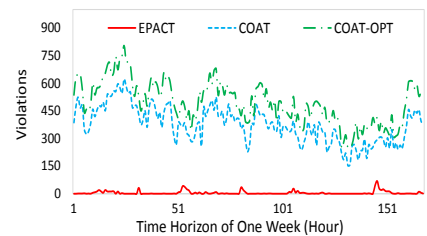Fig. 3: Server efficiency as UIPS/Watt under different core frequencies.

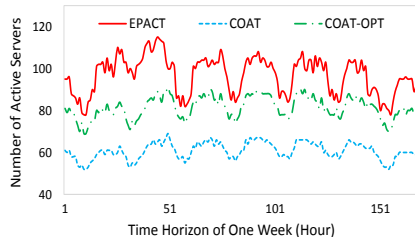Fig. 4: Violations per time slot for a time horizon of one week.

Fig. 5: Number of active servers for a time horizon of one week.
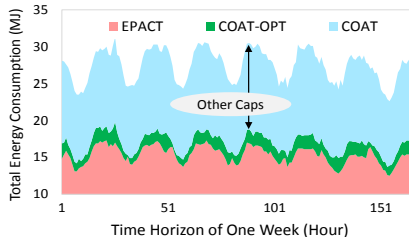

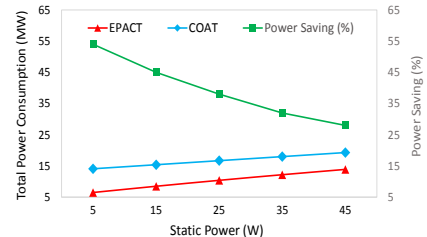Fig. 6: Energy consumed by data center for a time horizon of one week.

Fig. 7: Efficiency of proposed method under different static power.

when the static power (motherboard, fan, disk, etc.) increases from an efficient to a traditional power-hungry one. For higher static power consumption, optimal server frequency should be increased leading to higher CPU cap and lower number of active servers. These results prove that EPACT will be even more effective in future technologies, where static power is expected to decrease further.

VII. CONCLUSIONS

In this paper we presented an accurate power modelling for the proposed NTC servers based on the FD-SOI process technology. Then, we explored the existing energy vs. performance trade-offs when VMs with different CPU utilization and memory footprint characteristics are executed. We also evaluated the efficiency of our target virtualized applications on three different platforms: (i) x86, (ii) ARM-based Cavium ThunderX, and (iii) proposed NTC server. Finally, we proposed EPACT, a novel dynamic VM allocation method exploiting the given holistic knowledge of VMs characteristics and our new power model to increase the energy proportionality of next-generation NTC-based data centers while guaranteeing their QoS requirements. The proposed method has provided up to 45% energy savings when compared to conventional consolidation-based approach. Thus, our results demonstrate that the new NTC servers have created a completely new and promising (from an energy-efficiency viewpoint) research space on novel workload allocation techniques for next-generation data centers.

REFERENCES

[1] J. Koomey, "Growth in data center electricity use 2005 to 2010," Analytics Press, Oakland, CA, Tech. Rep., 2011.
[2] M. Shafique *et al.*, "Dark silicon as a challenge for hardware/software co-design," in *CODES+ISSS*, Oct 2014.
[3] A. Pahlevan *et al.*, "Towards near-threshold server processors," in *DATE*, 2016.
[4] D. Jacquet *et al.*, "A 3 ghz dual core processor ARM cortex TM -a9 in 28 nm UTBB FD-SOI CMOS with ultra-wide voltage range and energy efficiency optimization," *J. Solid-State Circuits*, 2014.

[5] N. Planes *et al.*, "Globalfoundries preps 12nm fdsoi process," in *http://www.eetimes.com/document.asp?doc_id=1330423*, 2016.
[6] L. Gwennap, "Fd-soi offers alternative to finfet," , Posted at https://www.globalfoundries.com/sites/default/files/fd-soi-offers-alternative-to-finfet.pdf, 2016.
[7] A. Beloglazov *et al.*, "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints," *IEEE TPDS*, vol. 24, no. 7, pp. 1366–1379, 2013.
[8] C. Delimitrou *et al.*, "Optimizing resource provisioning in shared cloud systems," Stanford University, Tech. Rep., 2014.
[9] U. R. Karpuzcu *et al.*, "Energysmart: Toward energy-efficient manycores for near-threshold computing," in *HPCA*, 2013.
[10] L. Gwennap, "Thunderx rattles server market," *Microprocessor Report*, vol. 29, no. 6, pp. 1–4, 2014.
[11] V. Laszewski *et al.*, "Power-aware scheduling of virtual machines in dvfs-enabled clusters," in *IEEE CLUSTER*, 2009.
[12] S. Esfandiarpoor *et al.*, "Structure-aware online virtual machine consolidation for datacenter energy improvement in cloud computing," *Computers & Electrical Eng.*, pp. 74–89, 2015.
[13] E. Pakbaznia *et al.*, "Minimizing data center cooling and server power costs," in *ISLPED*, 2009.
[14] D. Meisner *et al.*, "Power management of online data-intensive services," in *ISCA*, 2011.
[15] A. Pahlevan *et al.*, "Exploiting cpu-load and data correlations in multi-objective vm placement for geo-distributed data centers," in *DATE*, 2016.
[16] A. Verma *et al.*, "Server workload analysis for power minimization using consolidation," in *USENIX Annual Tech. Conf.*, 2009.
[17] J. Kim *et al.*, "Correlation-aware virtual machine allocation for energy-efficient datacenters," in *DATE*, 2013.
[18] X. Ruan *et al.*, "Performance-to-power ratio aware virtual machine (VM) allocation in energy-efficient clouds," in *IEEE CLUSTER*, 2015.
[19] S. K. Garg *et al.*, "Sla-based virtual machine management for heterogeneous workloads in a cloud datacenter," *Journal of Network and Computer Applications*, vol. 45, pp. 108 – 120, 2014.
[20] Micron, "4Gb: x4, x8, x16 DDR4 SDRAM features," https://www.micron.com/~/media/documents/products/data-sheet/dram/ddr4/4gb_ddr4_sdram.pdf, 2014.
[21] J. Wilkes, "More google cluster data," *Google research blog, November*, 2011.
[22] D. Bortolotti *et al.*, "User-space apis for dynamic power management in many-core armv8 computing nodes," in *HPCS*, 2016.
[23] D. Rossi *et al.*, "Energy-efficient near-threshold parallel computing: The pulpv2 cluster," *IEEE Micro*, vol. 37, no. 5, pp. 20–31, September 2017.
[24] G. E. P. Box *et al.*, "Time series analysis: Forecasting and control," *forth edition, Hoboken, NJ, USA: Wiley*, 2008.
[25] N. Binkert *et al.*, "The gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, 2011.