Realizing Complexity-Effective On-Chip Power Delivery for Many-Core Platforms by Exploiting Optimized Mapping

Mohammad Khavari Tavana*, Divya Pathak†,Mohammad Hossein Hajkazemi*, Maria Malik*,
Ioannis Savidis†, Houman Homayoun*
*Department of Electrical and Computer Engineering, George Mason University, Fairfax, Virginia, USA
†Department of Electrical and Computer Engineering, Drexel University, Philadelphia, Pennsylvania, USA
Email: *{mkhavari, mhajkaze, mmalik9, hhomayou}@gmu.edu †{divya.pathak, ioannis.savidis }@drexel.edu

***Abstract*—In the recent years, many-core platforms have emerged to boost performance while meeting tight power constraints. Per-core Dynamic Voltage and Frequency Scaling (DVFS) maximizes energy savings and meets the performance requirements of a given workload. Given a limited number of I/O pins and the need for finer control of voltage and frequency settings per core, there is a substantial cost in using off-chip voltage regulators. Consequently, there has been increased attention on the use of on-chip voltage regulators (OCVR) in many-core systems. However, integrating OCVRs comes at a cost of reduced power conversion efficiency (PCE) and increased complexity in the power delivery network and management of the OCVRs. In this paper, the effect of PCE on the thread-to-core mapping algorithm is investigated and the importance of the PCE-aware mapping scheme to optimize energy-efficiency is highlighted. Based on the results, up to 38% more energy savings is achieved as compared to PCE-agnostic algorithms. Moreover, the impact of core clustering granularity and process variation on the total efficiency of the system is explored. When relaxing the energy constraints by just 10%, an effective mapping reduces the complexity of the power delivery system by allowing the use of a significantly smaller number of voltage regulators, as compared to per-core OCVR. The results provided in the paper indicate an important opportunity for system and circuit co-design to implement energy-efficient and complexity-effective platforms for a target workload.**

Keywords—on-chip voltage regulator; workload mapping; energy-efficiency; power management; real-time systems

I. INTRODUCTION

The many-core paradigm has gained significant traction in the industry. Designed and fabricated processors that are categorized as many-core platforms include the Intel 80-core [1], the 167-core design by Truong et.al [2], the Intel single-chip cloud computer [3], and the Intel Xeon Phi. The novel paradigm has shifted the implementation of an integrated circuit from a few complex and over-provisioned cores to one incorporating a number of small and simple cores. Many-core platforms are most beneficial for current application demands, where multiple applications with multiple threads are running simultaneously. In addition, by taking advantage of parallelism, many-cores improve energy efficiency under a limited power budget. There are, however, significant design challenges that require addressing to fully realize the benefits of many-core platforms [5].

The power envelope is a major constraint to consider when designing many-core platforms. The best option to reduce the power dissipation is to use dynamic voltage and frequency scaling DVFS [1], [8]. Ideally in many-core platforms, fine grained power management is preferred, where the voltage and frequency of each core are individually controlled. However, as the number of cores increases with technology scaling, providing a voltage regulator (VR) per-core becomes costly. In addition, package pin counts are limited and a large PCB area is occupied by the voltage regulators, which constrains the use of many off-chip VRs.

Integrating VRs on-chip for chip multi-processors (CMPs) or many-core platforms, provides finer granularity in the management of power, improves the latency overhead of DVFS, and localizes noise. However, these benefits come at a cost. The area overhead of the on-chip voltage regulator (OCVR) is proportional to the required peak power. OCVRs typically provide between 0.5 and 5 A of current per millimeter square of area (the occupied area varies with the type of regulator and technology node) [28]. In addition, the area overhead of OCVRs is not likely to scale with technology scaling as most of the area occupied by the OCVRs is dedicated to the on-chip capacitors and inductors [27]. Furthermore, the power conversion efficiency (PCE) of OCVRs is lower than off-chip VRs [6], [7], [27]. Nonetheless, architecture and system level optimization techniques have been proposed to exploit the fast DVFS response of OCVRs (in the range of tens of nanoseconds) to further improve the energy efficiency as compared to off-chip VRs (DVFS response of a few milliseconds) [6], [13], [16], [27]. The architectural exploration to use fast OCVRs is investigated by Kim et al. [6]. Substantial energy saving is possible when using per-core DVFS as shown in [16]. The trade-offs between on-chip and off-chip VRs is described in [27]. The high efficiency of off-chip VRs and low latency of on-chip VRs is exploited through application migration in multi-programmed workloads [27]. On-chip distributed VRs to implement both fast core resizing and voltage scaling are exploited in [13] to increase energy-efficiency.

Another significant challenge for many-core platforms is the process variation (PV) that is exacerbated for each progressive technology generation. The challenge of PV is addressed in this paper as circuits are evaluated with different levels of applied PV. PV in conjunction with the high number of cores in many-core platforms leads to a fluctuation in the maximum frequency (f_{max}) and leakage power dissipation of each core [1], [4], [10].

978-1-4673-7166-7/15/$31.00 ©2015 IEEE

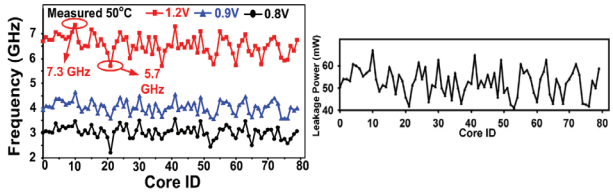

Fig 1: Core frequency and leakage power dissipation of 80 core Intel processor [1].

The core frequency and leakage power dissipation for an 80-core Intel chip [1] is shown in Fig 1. At the maximum supply voltage, there is 28% variation between the fastest and slowest core. The variation in frequency increases to 59% at the minimum voltage (0.8V). On the other hand, the variation in leakage power reaches 63%. Many-core processors experience higher core-to-core frequency variation in the presence of DVFS [1], [8], [10]. A PV-aware algorithm based on linear programming for mapping and power management of CMPs was proposed by Teodorescu and Torrellas [18]. The throughput of multi-programmed workloads is maximized given a fixed power budget. Dynamic thread mapping for multi-threaded workloads is proposed in [9] to reduce application latency. Dynamic application behavior in the presence of PV is considered and threads are mapped to the best suited cores. Thread count determination, mapping, and power management to minimize energy consumption are proposed in [10] for multi-threaded embedded applications. The impact of the PCE of the VR has been overlooked in thread-to-core mapping algorithms. Neglecting the PCE in the optimization procedure, leads to sub-optimal solutions that impact the energy efficiency of the system.

A cost-effective solution for per-core voltage domains by exploiting on-chip low-drop-output (LDO) VRs for CMPs is proposed by Sinker et al. [8]. However, given that the largest area of OCVRs is occupied by on-chip inductors and capacitors that do not sufficiently scale with each technology generation, it is difficult to integrate many OCVRs. Alternatively, integrating high quality on-chip inductors is also a challenge due to physical constraints caused by technology scaling [8].

In this paper, the design of the OCVRs is considered for many-core platforms while accounting for system and algorithm level techniques to reduce the complexity of OCVRs in the early stage of a design. In addition, the effect of the OCVR efficiency and the topology of the power delivery network on the thread mapping algorithm is explored.

Results indicate that there is a substantial opportunity for system and circuit co-design to realize cost-effective power delivery for many-core platforms. The main contributions of this paper are as follows:

- A system level optimization technique is developed which is aware of the PCE of the OCVRs and intra-chip process variations. The simulated results indicate that by clustering cores to be served by a single OCVR, approximately the same energy efficiency is obtained as per-core DVFS, thus significantly reducing the OCVR area and power overhead.

TABLE I CORE V/F PAIRS AND OCVR PCE SPECIFICATION FOR THE MANY-CORE SYSTEM.

Voltage (V)	Frequency(GHz)	PCE (%)
1.0	2.4	90
0.9	2.2	81
0.8	2.0	72
0.7	1.7	63

- The OCVR PCE-aware mapping algorithm maximizes the energy savings in many-core platforms. Up to 38% improvement in energy efficiency is obtained when accounting for the PCE of the state-of-the-art on-chip LDO regulator in the mapping algorithm.
- A simulated annealing-based mapping algorithm is proposed where the number of threads, mapping, and voltage/frequency scaling are simultaneously accounted for to optimize energy efficiency.

The rest of the paper is organized as follows: Design considerations for OCVRs in many core systems are provided in Section II. Models and assumptions for the voltage regulator, system, and application are described in Section III. The mapping algorithm is described in Section IV. The simulation framework, evaluations, and results are given in Section V. Finally, conclusions are offered in Section VI.

II. ON-CHIP VOLTAGE REGULATION

In the post-Dennard era, the lower noise margin in high performance integrated circuits (IC) requires precise voltage delivery with low susceptibility to noise. With single core computing, simplified power supply schemes implemented off-chip voltage regulator to serve the digital blocks. However, with the advent of multi-core and many-core computing, off-chip regulators were inefficient as significant power loss occurs when a single voltage regulator serves all the cores [27], [6]. Using multiple off-chip voltage regulators to serve individual cores increases the occupied PCB area significantly when the VRs are implemented using passive components. The DVFS latency offered by off-chip voltage regulators is not sufficient to meet the dynamic variation in processing core demand leading to further power loss.

Shifting voltage regulators on-chip to locally serve each voltage domain at per-core granularity solves many of the problems mentioned above. The *IR* drop is reduced due to two reasons. First, the shorter interconnect length between the OCVR and the load circuit produces a lower resistance. Second, a higher voltage is brought on chip with a corresponding reduction in the current (power remains constant). The *Ldi/dt* noise is localized and reduced due to the smaller inductance of on-chip interconnects as compared to the inductance of the large wire bonds or ball grid arrays. The OCVRs offer faster DVFS, generally two to three orders of magnitude higher than that of off-chip VRs. The integrated voltage regulator module (iVRM) in the IBM Power8 [14] and the fully integrated voltage regulator (FIVR) in the Intel Haswell [15] processors are two examples of OCVR integration in high performance multi-core systems. In this work, the OCVR is modeled similar to the iVRM. The iVRM architecture exploits a low drop-out (LDO) regulator which is a special class of series linear DC-DC regulators. Series regulators offer superior output voltage

regulation as compared to shunt regulators and utilize a smaller area as passive components are not included in the design.

The total footprint of an LDO is determined by the pass element width. The size of an LDO ranges from 60 x 60 μm^2 for a load current of 0.5 A to 150 x 150 μm^2 for a load current of 3.5 A [17]. The low output impedance of an LDO allows for faster load regulation and reduced voltage noise at the output node. The small footprint, low cost, low noise, and fast response to load current transients offered by an LDO make it a suitable candidate for on-chip voltage regulation.

The increasing power density of cores requires efficient OCVRs with small area, high PCE, and fast voltage changes. The criteria for selection of an optimum OCVR topology in many-core systems with small cores are more complex. The increasing number of voltage domains and per-core DVFS in many-core systems require careful planning to distribute OCVRs to facilitate point of load power delivery that maximize the PCE. In many-core architectures with hundreds of cores, providing each core with a dedicated OCVR so as to facilitate per-core DVFS is an area intensive and expensive solution. In addition, managing hundreds of OCVRs requires the Power Management Unit (PMU) to execute sophisticated algorithms. An optimum topology, number and placement of the OCVRs requires a system level study where the objective is to maximize the system energy efficiency and simultaneously minimize the power supply noise, parasitics, and area overhead.

III. System, Circuit, and Application Models

A. Voltage regulator model

A many-core system is modeled in a 22 nm technology with a per core peak power (P_{Peak}) requirement of 1.5 W. At a power supply voltage (V_{DD}) of 1 V, an LDO similar to iVRM offers a peak PCE of 90%. With DVFS, the PCE will drop to 63% for a V_{DD} of 0.7 V. Previous work completed by Zeng et.al [21], determined that the reduction in PCE is negligible as the number of LDOs supplying current to a Power Distribution Network (PDN) is increased. The DVFS pairs and the corresponding PCE for the LDO modeled on the iVRM for the many-core system are shown in Table I. The chosen OCVR model offers the flexibility to linearly increase the number of LDOs that serve an increasing number of small cores each with a peak power requirement equal to P_{Peak}. The simulation model adopted for the many-core system is explained in Section III-B.

B. Many-core system model

The modeled many-core platforms include varying number of cores (N_{core}={32, 64, 128, 256}), with each core consuming a peak power of P_{Peak}. Three cases each with different complexity and capability are shown In Fig. 2.

- *Per-core voltage regulator:* Each core has a dedicated OCVR. The voltage and frequency for each core is set individually to exploit core-to-core process variation. Even though this configuration is flexible, the power delivery system and OCVR management are complex and the OCVRs impose a huge area overhead. Since the Power Management Unit (PMU) directly controls the OCVRs, turning them on or off, a power gating circuit (PGC) per core is not needed (Fig. 2a).
- *Per-cluster voltage regulator with separate PLL (PCSP):* Each OCVR serves a cluster of cores. Voltage is managed at the cluster level whereas the frequency is set individually per-core to exploit the process variation within the clusters. Per-core frequency control necessitates the use of a separate PLL for each core to generate the clock signal (Fig. 2b). In addition, a synchronization mechanism such as using FIFO buffers is required between the different frequency domains [18]. This configuration exploits the frequency heterogeneity among cores and was implemented in the AMD Quad-Core Opteron [19].
- *Per-cluster voltage regulator with common PLL (PCCP)*: Each OCVR serves a cluster of cores and one PLL generates the clock signal for all the cores within a cluster (Fig. 2c). Both voltage and frequency is managed at the cluster level, however the frequency is set by the weakest core within a cluster. The process variation is therefore localized within the clusters and the platform exploits the process variation among the clusters.

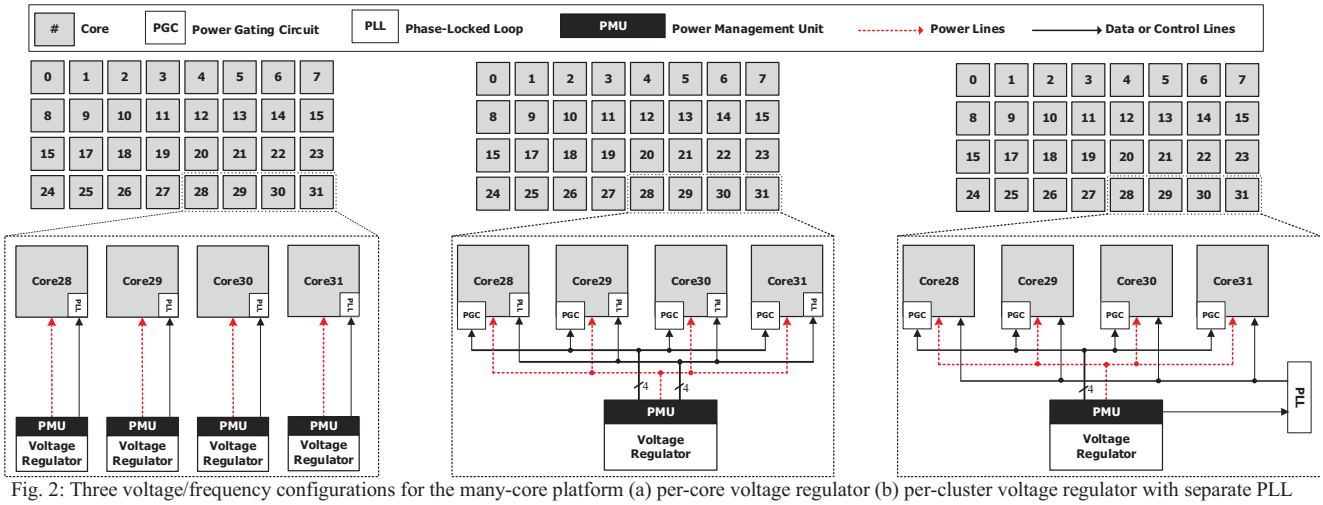

Fig. 2: Three voltage/frequency configurations for the many-core platform (a) per-core voltage regulator (b) per-cluster voltage regulator with separate PLL (PCSP), and (c) per-cluster voltage regulator with common PLL (PCCP).

Implementing PCCP reduces the complexity of the PCSP configuration by eliminating the use of separate PLL and synchronization circuit.

C. Application Model

Two types of applications are simulated. First, workloads with real-time applications are considered in which each application must be finished before a predefined deadline (D). For this class of workloads, a thread-to-core mapping is applied which results in minimum energy consumption without violating the deadline. Second, workloads with high performance applications are considered. There is no deadline for high performance workloads, and a fast response time is desired instead. Therefore, for this class of workloads, the goal is to minimize the energy-delay-squared product (ED^2P) [13], [18], [20]. All the benchmarks used to create the workloads are multi-threaded, which represents of common applications for many-core platforms.

D. Process Variation Model

Due to the imperfections in the manufacturing process in the sub-nanometer nodes, the maximum frequency and dissipated leakage power varies from core to core. The impact of process variation depends on many parameters including system design and manufacturing process technology [24]. VARIUS [24] is used to model variability and study within die variation (WID), including systematic and random variations. The chip surface is modeled as a grid consisting of $N_{grid} \times N_{grid}$ cells, each of which represents the value of process parameter (i.e., V_{th} and L_{eff}). Both random and systematic variations are modeled as normal distributions with μ (mean) and σ (standard deviation) parameters. In addition, the process parameter at two different grid cells at distance r is correlated with a coefficient factor of $\rho(r)$ that reduces with increasing distance as expressed in (1).

$$\rho(r) = \begin{cases} 1 - \frac{3r}{2\phi} + \frac{r^3}{2\phi^3} & r \le \phi \\ 0 & otherwise \end{cases} \quad (1)$$

The correlation range ϕ is expressed as a fraction of the chip length [24]. A larger ϕ indicates that large portions of the chip are correlated with each other.

For the developed model, the chip surface is divided into 100×100 grid cells (N_{grid} =100). The standard deviation of the process variations is set to 10% of the nominal process parameter [10], [29], [30]. The correlation range parameter ϕ for systematic variation is set to 0.5, which is a realistic assumption for a typical microprocessor die [24].

E. Power, Energy and ED^2P Models

1) Optimization objective for real-time applications: The total power consumption of core i at the voltage of V_i and a frequency of f_i is expressed as the aggregation of the dynamic (P_D) and leakage (P_S) power:

$$P_i = \frac{P_D(V_i, f_i) + P_S(V_i)}{PCE(V_i)} = \frac{C_{eff} V_i^2 f_i + V_i I_i^{Leakage}}{PCE(V_i)} \quad (2)$$

where C_{eff} is the effective capacitance of the core while executing a given application, f_i is the operational frequency, $I_i^{Leakage}$ is the average leakage current, and PCE is the power conversion efficiency of the OCVR at a voltage of V_i (see Table I).

For real-time applications, the goal is to minimize the energy consumption of the system while assuring that all applications meet execution deadlines. According to the power model, the energy consumption of application j while executing ℓ number of threads is written as:

$$E_i = \frac{P_D(V_i, f_i) + P_S(V_i)}{PCE(V_i)} T_j(f_i, \ell) \quad (3)$$

where $T_j(f_i, \ell)$ is the normalized execution time at the given frequency f_i with ℓ number of threads. Therefore, the optimization objective of the many-core system with N_{core} number of cores, provided that all the applications (N_{app}) in the workload meet specified deadlines (D_j) is desired as:

$$Min \sum_{i=1}^{N_{core}} E_i = Min \sum_{i=1}^{N_{core}} \frac{P_D(V_i, f_i) + P_S(V_i)}{PCE(V_i)} T_j(f_i, \ell)$$
$$subject\ to: for\ \ all\ j \in N_{app}\ Max\left\{T_j(f_i, \ell)\right\} \le D_j \quad (4)$$

Note that cores on which no application is running are power gated by the PMU and are therefore consuming no energy. In addition, from the view of the application, the slowest thread determines the execution time of the application. Therefore, the maximum execution time of application threads must not violate the set deadline.

2) Optimization objective for high-performance applications: The ED^2P metric is considered as the objective function for high performance applications [13], [18], [20]. The ED^2P metric emphasizes performance and only sacrifices performance if the energy benefit is significant. Since, ED^2P gives higher priority to the performance over the power, it is a more appropriate metric for high-performance systems. The objective function for one multi-threaded high-performance application is expressed as follows:

$$Min(ED^2P_j) = Min\left\{\left(\sum_{i \in N_{core}} E_i\right) \underset{j \in N_{app}}{Max}\left([T_j(f_i, \ell)]\right)^2\right\} \quad (5)$$

The first component of (5) is the summation of the energy consumption of the cores on which the thread for the application is running. The second component of the objective function is the maximum execution time of multiple threads (ℓ) of application j in the many-core platform. Due to process variation and heterogeneity of the core frequency, each thread of the application finishes with different times [10]. The same weight for all applications is considered and the normalized ED^2P is used to prevent an application with large execution time biasing the results. All the execution times are therefore normalized to the case where the applications are executed serially at the maximum frequency. The objective function for the entire many-core platform is to minimize the geometric mean of ED^2P for all applications in the workload [20], and is written as:

$$Min\left(\prod_{j=1}^{N_{app}} ED^2P_j\right)^{1/N_{app}} =$$
$$Min\left\{\left(\prod_{j=1}^{N_{app}}\left(\left(\sum_{i \in N_{core}} E_i\right) \underset{j \in N_{app}}{Max}\left([T_j(f_i, \ell)]\right)^2\right)\right)^{1/N_{app}}\right\} \quad (6)$$

The mapping algorithm explained in the following section accounts for the OCVR PCE. In addition, the algorithm uses the developed formulae in Section III to determine the number of threads for each application, map them to the cores, and select the optimum voltage and frequency.

IV. MAPPING ALGORITHM

Similar to the mapping problem in heterogeneous hardware, mapping a workload on a many-core platform with power and frequency variation is NP-hard [25]. Considering workloads with multi-threaded applications further exacerbates the complexity of the problem. One of the popular solutions to effectively map applications in many-core platforms is Simulated Annealing (SA) [10], [11]. In this work, the simulated annealing-based mapping framework for multi-threaded workloads is used. The mapping algorithm attempts to move out from local optima by probabilistically allowing for acceptance of poorer solutions [31]. Three different configurations with various number of OCVRs are investigated. The results show that the effective mapping alleviates the complexity of the underlying power delivery system by allowing a lower number of OCVRs in the design.

The mapping algorithm is presented in Algorithm 1. In line 1 of the algorithm, some required parameters including *cooling rate*, *initial temperature* (T_0), and *Initial iteration* are initialized. Initial iteration is set to 500 and indicates the number of repetitions for each temperature. The initial temperature is set to 60, and is chosen large enough such that the algorithm accepts most of the worst solutions, effectively exploring the entire search space. The cooling rate is set to 0.9 and is chosen slow enough to preclude the mapping algorithm from selecting a local minimum. The termination condition (line 3) for selecting the final solution executes when the current solution remains unchanged for many iterations. A bound for the minimum temperature is also set, which stops the algorithm when crossed. The bound is small enough (0.005) such that the solution remains frozen for most of the time.

The generation of new solutions (line 4) is the fundamental part of the simulated annealing algorithm for the different cases. The algorithm begins with a random feasible mapping (line 2) as an initial solution and progresses with new heuristic movements from the current state. The new generated solution is introduced by small changes in the current state. The movements are minimal alternations to the previous solution, allowing for a vast search of the solution space. In other words, a solution which violates the deadline constraints is not a feasible solution in the search space, however, for high performance workloads every solution is a possible one. The heuristic movements attempt to change the number of threads and thread-to-core mapping simultaneously for each application. The heuristic movements are as follows:

- *Increasing/decreasing number of threads.* This movement selects an application randomly and increases/decreases the number of threads with respect to the allowable thread count for a particular application.
- *Intra-cluster task swapping.* Two random cores are selected within a cluster, and the corresponding threads are swapped. If one of the selected cores is idle, the simple thread migration is performed.
- *Inter-cluster task swapping.* Two random cores are selected from two different clusters, and the corresponding threads are swapped. Similarly if one of the selected cores is idle, the simple thread migration is performed.
- *Cluster swapping.* Two random clusters are selected and all the threads are swapped between clusters.
- *Random movement.* Although the former heuristic movements are systematic, including randomness improves the exploration of search space and avoids selecting a local optima.

After performing heuristic movements, the energy efficiency metric (i.e., energy for real-time and ED^2P for high-performance workloads) of the entire system for all the possible voltages is calculated (line 5). A voltage for a core or a cluster which results in the best energy-efficiency is selected.

We define Δ as the difference between the energy efficiency metric of the new mapping and current mapping (line 6). If Δ is negative, the generated solution (S') is the more energy efficient solution and is used to replace the current solution (line 7). Otherwise, the solution is accepted with probability of $e^{-\Delta/T}$ (line 8), where T is the current temperature. The larger the value of T, the higher the probability of accepting a worse solution. By reducing T with respect to the cooling rate, the probability of accepting the worst solution is reduced (line 10). In addition, when the temperature drops below 1, the number of iterations increases to better investigate the search space (line 11).

***Algorithm 1**: Simulated Annealing-based Mapping Algorithm*

Inputs:
-Workload, dynamic power and execution time of each task with corresponding number of threads. System architecture including number of cores, clustering configuration, maximum frequency of each core for a given voltage and leakage power.

Outputs:
-Thread-to-core mapping, Number of threads for each benchmark, voltage and frequency for clusters/cores.

begin
1: initialize T_0, *cooling rate* and *initial iteration*
2: S = generate a random and feasible solution
3: ***While*** *stop condition* is not met:
4: S' = generate feasible solution from current solution (S) by applying heuristic movements
5: find voltage and frequency for clusters (cores) which leads to the maximum energy efficiency (based on (4) and (6))
6: Δ = energy(S') - energy(S)
7: ***if*** $\Delta \leq 0$ then
$S = S'$
8: ***else***
$S = S'$ with probability of $e^{-\Delta/T}$
9: ***if*** enough iterations have completed:
10: reduce *Temperature* with respect to *cooling rate*
11: multiply *iteration* parameter by $2^{log10(1/T)}$
12: ***end while***
end

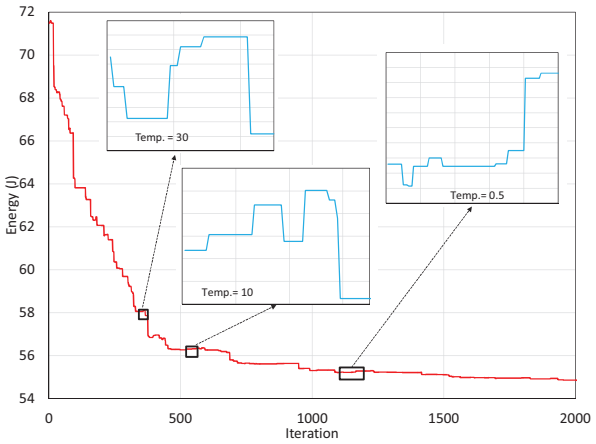

Fig. 3: Visualization of the simulated annealing algorithm searching for the best solution the problem space.

The implemented simulated annealing-based mapping is effectively used for both energy efficiency metrics ((4) and (6)). Note that different simulated annealing algorithms for solving various optimization problems are available [31]. In this work, a new mapping algorithm or even an improvement in the performance of the existing simulated annealing algorithm are not proposed, but rather the meta-heuristic algorithm is used as a tool to effectively solve the mapping problem in many-core platforms with heterogeneous power and frequencies. To better understand the execution of the algorithm, the first 2000 iterations of the mapping procedure of a workload with 20 applications on a 64-core platform are shown in Fig. 3. With temperature reduction, the probability of accepting the worse solution is reduced, and eventually leads to freezing of a mapping solution. Increasing the number of cores increases the search space exponentially. The algorithm is executed off-line and cannot guarantee finding a global optima as the problem is NP-hard. The initial parameters of the algorithm as well as the number of iterations affect the final solution. However, the parameters are selected conservatively to allow for the algorithm to find a good and acceptable solution, as the main objective is the investigation of the effects of algorithm and system level optimizations on the underlying power delivery and OCVRs.

V. SIMULATIONS AND RESULTS

An overview of the simulation methodology is shown in Fig. 4. To generate high-performance and real-time workloads, ten applications from the Parsec benchmark suite [22] and ten from the ParMibench benchmark suite [26] are executed on an Atom architecture at four different frequencies (see Table I). The Parsec results are used to generate 300 random high-performance workloads and the ParMibench results are used to generate 300 random real-time workloads. For real-time applications the deadline is assumed to be equal to the serial execution time of each application at nominal frequency. Each workload contains {10, 20, 30, 40, 50, 60} multi-threaded applications. The performance of each application is carefully measured using the Linux "perf" tool [12], and "cpufrequtils" [23] is used for scaling the frequency of the cores. The speedup of various applications at maximum frequency are reported in Tables II and III for, respectively, the Parsec and ParMibench

TABLE II SPEEDUP OF PARSEC BENCHMARKS ON THE INTEL ATOM PROCESSOR WITH DIFFERENT THREADS AT THE MAXIMUM FREQUENCY

Speedup with different number of threads							
Threads	**2**	**3**	**4**	**5**	**6**	**7**	**8**
dedup	1.35	1.37	1.45	1.33	1.34	1.44	1.60
streamcluster	1.92	2.50	2.75	2.64	2.83	2.60	2.55
blackscholes	1.83	2.52	3.12	3.63	4.08	4.46	4.77
bodytrack	1.89	2.69	3.41	4.07	4.66	5.15	5.62
ferret	1.90	2.35	2.24	2.26	2.21	2.22	2.25
fluidanimate	1.74	NA*	2.75	NA	NA	NA	3.88
freqmine	1.71	2.24	2.58	2.91	3.06	3.14	3.20
swaptions	2.00	2.90	3.99	3.99	4.55	6.35	7.94
vips	1.97	2.89	3.79	4.61	5.69	6.27	7.02
x264	1.92	2.84	3.72	4.42	5.23	5.74	5.45

Table III SPEEDUP OF PARMIBENCH ON THE INTEL ATOM PROCESSOR WITH DIFFERENT THREADS AT THE MAXIMUM FREQUENCY

Speedup with different number of threads							
Threads	**2**	**3**	**4**	**5**	**6**	**7**	**8**
dijkstra	1.31	NA*	1.52	NA	1.46	NA	1.57
susan smooth	1.98	NA	3.82	NA	5.15	NA	7.48
susan edge	1.89	NA	3.25	NA	3.94	NA	4.65
susan corner	1.77	NA	2.77	NA	3.25	NA	3.89
patricia	1.25	NA	1.61	NA	1.80	NA	1.79
sha	1.59	1.76	1.97	1.87	1.53	1.81	2.59
str search 1	1.93	2.83	3.56	4.32	5.01	5.66	6.29
str search 2	1.99	2.99	3.97	4.98	5.95	6.93	7.89
square root	1.99	2.99	3.97	4.98	5.95	6.93	7.89
cubic eq. solver	2.00	3.00	4.00	5.00	5.98	6.98	7.96

*Execution was not possible due to the type of application

benchmarks. Since process variation is considered in the simulations, the execution time is interpolated for frequencies which are not measured on the Atom. Note that the frequency binning was performed in 50 MHz steps for the simulations. Consumption of random dynamic power in also included in the simulation for each task and is uniformly distributed from 500 mW to 1.5 W.

The many-core platform is modeled with a various number of cores (N_{core}={32, 64, 128, 256}). Three different configurations for clustering are explored (see Fig. 2). An LDO similar to the one employed in the IBM iVRM Power8 [14] is

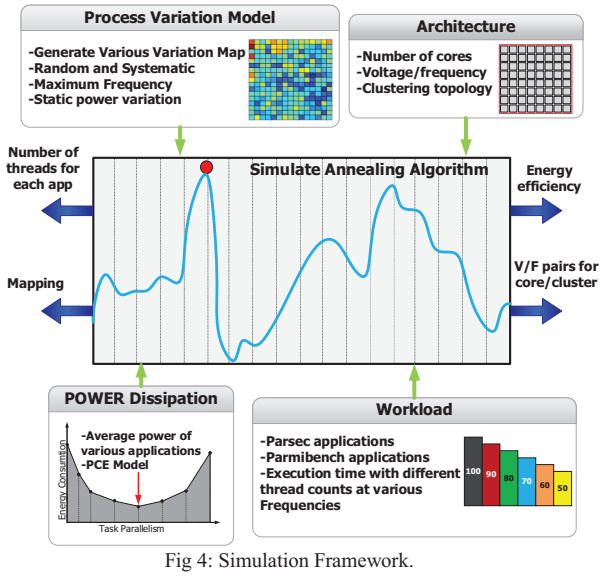

Fig 4: Simulation Framework.

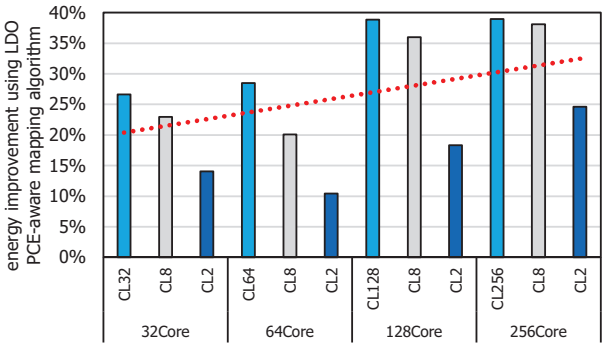

Fig 5: Impact of the PCE-aware mapping algorithm on energy savings with various number of cores and clusters (CL#).

modeled for the power delivery. The objective of the simulated annealing framework is to map all the threads to cores that optimize the energy-efficiency metric for each type of workload. The output of the implemented simulated annealing framework is a mapping of the threads, the total number of required threads for each application, and the voltage/frequency selection.

A. PCE-aware mapping

The PCE varies at different voltage levels. The algorithm must account for the variation in the PCE as it is one of the main sources of inefficiency when effectively mapping the workload. Considering the PCE of the VR significantly changes the effectiveness of the mapping. A many-core platform running a real-time workload with LDO OCVRs is modeled to show the effect of PCE-aware mapping on the energy efficiency. Two different cases are simulated: (1) The algorithm considers the maximum PCE (90% for LDO) at all the voltage levels, and (2) the algorithm is aware of the variation in the PCE at different voltage levels (see Table I). The objective of the simulation is to show that neglecting the PCE produces a sub-optimal energy-efficient mapping. We assume a per-core configuration, a PCSP with 8 OCVRs (8 clusters), and PCSP with two OCVRs (2 clusters). The reported results are the average of all simulations with 10 to 60 applications in the simulated workloads. 10% to 38% more energy is consumed when the mapping algorithm is un-aware of the variation in PCE, as shown in Fig. 5. With an increasing number of cores in the platform, the mapping algorithm utilizes more cores and assigns more threads per application. Consequently, due to the increase in slack time, applications have more potential to execute at lower

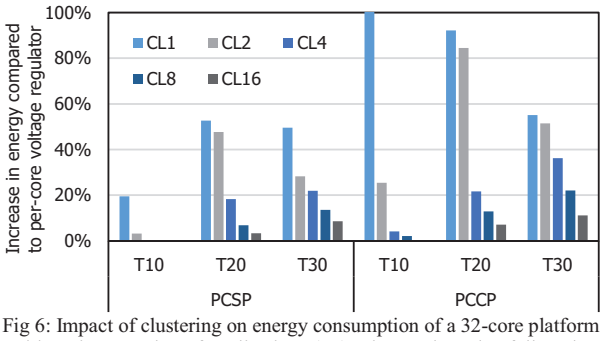

Fig 6: Impact of clustering on energy consumption of a 32-core platform with various number of applications (T#). The numbers that follow the CL notation indicate the number of clusters.

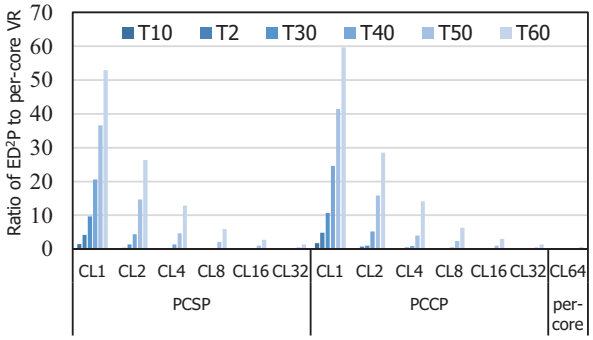

Fig 7: Impact of clustering on ED^2P of a 64-core platform with various number of applications (T#). The numbers that follow the CL notation indicate the number of clusters.

voltage/frequency. Therefore, when the number of cores increases, PCE-unaware mapping excessively reduces the voltage and increases the gap in energy consumption as compared to the PCE-aware mapping. The trend line also indicates the same conclusion. The same argument is also valid when the number of clusters (number of OCVRs) increases.

B. Impact of clustering

The three clustering configurations are explored with a varying number of multi-threaded applications for a 32-core platform and the resulting energy consumption is shown in Fig 6. The energy consumption is compared to the case with a per-core voltage regulator. Since in PCSP, the mapping algorithm is able to exploit process variation within each cluster, in all cases it outperforms the PCCP configuration. When the number of clusters is small (i.e. 1 or 2), the increase in energy consumption is large, particularly for the PCCP scheme.

As listed in Table IV, when relaxing the energy constraint by just 10%, an effective mapping reduces the complexity of the power delivery system by allowing the use of a significantly smaller number of OCVRs as compared to per-core OCVR. Note that the number of OCVRs increases as the maximum number of multi-threaded applications that simultaneously run on the many-core increases. For instance, four OCVRs in a 128-core platform are sufficient to satisfy the design constraint if there are up to 40 applications in a workload.

For high performance applications, the geometric mean of the ED^2P is optimized. The ED^2P metric for a different number of applications and clustering configurations for a 64-core platform is shown in Fig 7. All the values are normalized to the per-core OCVR configuration. The ED^2P is highly sensitive to the number of clusters (OCVRs) as well as the number of applications in the workload. Therefore, reducing the number of

TABLE IV EFFICIENT NUMBER OF OCVRS WITH DIFFERENT TASKS IN WORKLOAD

# of Apps	10	20	30	40	50	60
CL32-PCSP	2	8	16	NA	NA	NA
CL32-PCCP	4	16	32	NA	NA	NA
CL64- PCSP	2	4	4	8	32	32
CL64- PCCP	4	4	8	16	32	32
CL128- PCSP	2	2	4	4	8	16
CL128- PCCP	8	8	8	8	8	16
CL256- PCSP	2	4	4	4	8	8
CL256- PCCP	4	4	16	16	16	16

OCVRs excessively in a design results in a platform with poor efficiency. As expected, the PCSP configuration outperformers the PCCP. However, if the number of applications in the workload is less than half of the number of cores in a 64-core system (30, 20, and 10), then the small number of OCVRs results in good efficiency.

VI. Conclusion

DVFS is an effective technique to reduce the power consumption of a system. However, in many-core systems, while applying DVFS per-core leads to a maximum energy savings, there is an increased cost in power delivery complexity and management of the on-chip voltage regulators.

In this paper, the impact of the state-of-the-art on-chip voltage regulation topology on energy efficiency of a many-core system is evaluated. Simulated annealing-based mapping is used for system and algorithm level optimization to reduce the complexity of the power delivery system. The efficiency is further enhanced by adapting the mapping algorithm based on the underlying power delivery system design specifications. In addition, the importance of the PCE-aware mapping scheme to optimize energy-efficiency is highlighted.

Results indicate that when the energy constraints are relaxed by just 10%, an effective mapping reduces the complexity of the power delivery by allowing for the use of a significantly smaller number of voltage regulators, as compared to per-core voltage regulation.

References

[1] S. Dighe, et al., "Within-die variation-aware dynamic-voltage-frequency-scaling with optimal core allocation and thread hopping for the 80-core teraflops processor," *IEEE Journal of Solid-State Circuits*, Vol 46, No. 1, pp. 184-193, 2011.

[2] N. Truong, et al., "A 167-processor computational platform in 65 nm CMOS," *IEEE Journal of Solid-State Circuits*, Vol. 44, No. 4, pp. 1130-1144, 2009.

[3] P. Salihundam, et al., "A 2 tb/s 64 mesh network for a single-chip cloud computer with dvfs in 45 nm cmos," *IEEE Journal of Solid-State Circuits*, Vol. 46, No. 4, pp. 757-766, 2011.

[4] K. Rangan, M. Powell, G. Wei, and D. Brooks, "Achieving uniform performance and maximizing throughput in the presence of heterogeneity," *IEEE 17th International Symposium on High Performance Computer Architecture*, pp. 3-14, 2011.

[5] S. Borkar, "Thousand core chips: a technology perspective," *In Proceedings of the 44th annual Design Automation Conference*, pp. 746-749, 2007.

[6] W. Kim, M. Gupta, G. Wei, and D. Brooks, "System level analysis of fast, per-core DVFS using on-chip switching regulators," *IEEE 14th International Symposium on High Performance Computer Architecture*, pp. 123–34, 2008.

[7] W. Kim, D. Brooks, and G. Wei, "A fully-integrated 3-level dc-dc converter for nanosecond-scale dvfs," *IEEE Journal of Solid-State Circuits*, Vol. 47, No.1, pp. 206–19, 2012.

[8] A. Sinkar, H. R. Ghasemi, M. J. Schulte, U. Karpuzcu, and N. K. Kim, "Low-Cost Per-Core Voltage Domain Support for Power-Constrained High-Performance Processors," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 22, No. 4, pp. 747-758, 2014.

[9] S. Hong, S. Narayanan, M. Kandemir, and Ö. Özturk, "Process variation aware thread mapping for chip multiprocessors," *In Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 821-826, 2009.

[10] M. K. Tavana, A. Kulkarni, A. Rahimi, T. Mohsenin, and H Homayoun, "Energy-efficient mapping of biomedical applications on domain-specific accelerator under process variation," *In Proceedings of the international symposium on Low power electronics and design*, pp. 275-278, 2014.

[11] A. K. Singh, M. Shafique, A. Kumar, A., and J. Henkel, "Mapping on multi/many-core systems: survey of current and emerging trends," *In Proceedings of the 50th Annual Design Automation Conference*, pp. 1-10, 2013.

[12] available at: https://perf.wiki.kernel.org/index.php/Main_Page

[13] M. K. Tavana, M. Hajkazemi, D. Pathak, I. Savidis, and H. Homayoun, "ElasticCore: Enabling Dynamic Heterogeneity With Joint Core and Voltage/Frequency Scaling", *In Proceedings of the 52th Annual Design Automation Conference*, pp. 1-6, 2015.

[14] E.J. Fluhr, et al. "The 12-Core POWER8™ Processor with 7.6 Tb/s IO Bandwidth, Integrated Voltage Regulation, and Resonant Clocking," *IEEE Journal of Solid-State Circuits*, Vol. 50, No. 1, pp. 1-12, 2015.

[15] P. Hammarlund, et al., "Haswell: The Fourth-Generation Intel Core Processor," *IEEE Micro*, Vol. 34, No. 2, pp. 6–20, 2014.

[16] S. Eyerman and L. Eeckhout. "Fine-grained DVFS using on-chip regulators," *ACM Transactions on Architecture and Code Optimization*, Vol. 8, No. 1, 2011.

[17] I. Vaisband and E.G. Friedman, "Heterogeneous Methodology for Energy Efficient Distribution of On-Chip Power Supplies," *IEEE Transactions on Power Electronics*, Vol. 28, No. 9, pp. 4267–4280, 2013.

[18] R. Teodorescu and J. Torrellas, "Variation-Aware Application Scheduling and Power Management for Chip Multiprocessors," *In ACM SIGARCH Computer Architecture News*, Vol. 36, No. 3, pp. 363-374, 2008.

[19] J. Dorsey, et al., "An integrated quadcore Opteron processor," *ISSCC*, Session 5, Microprocessors, pp. 102-103, 2007.

[20] D. Brooks, et al., "Power-aware microarchitecture: Design and modeling challenges for next-generation microprocessors," *IEEE Micro*, Vol. 20, No.6, pp. 26-44, 2000.

[21] Z. Zeng, X. Ye, Z. Feng, and P. Li "Tradeoff analysis and optimization of power delivery networks with on-chip voltage regulation," *In Proceedings of the 47th Annual Design Automation Conference*, pp. 831-836, 2010.

[22] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," *In Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, pp. 72-81, 2008.

[23] https://wiki.archlinux.org/index.php/CPU_frequency_scaling

[24] S. Sarangi, et al., "VARIUS: A model of process variation and resulting timing errors for microarchitects," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 21, No.1, pp. 3-13, 2008.

[25] A. Schranzhofer, J. J. Chen, and L. Thiele, "Dynamic power-aware mapping of applications onto heterogeneous mpsoc platforms," *IEEE Transactions on Industrial Informatics*, Vol. 6, No.4, pp. 692-707, 2010.

[26] S. Iqbal, Y. Liang, and H. Grahn, "ParMiBench-An Open-Source Benchmark for Embedded Multiprocessor Systems," *IEEE Computer Architecture Letters*, Vol. 9, No. 2, pp 45-48, 2010.

[27] G. Yan, et al., "AgileRegulator: A Hybrid Voltage Regulator Scheme Redeeming Dark Silicon for Power Efficiency in a Multicore Architecture," *IEEE 18th International Symposium on High Performance Computer Architecture*, pp. 1–12, 2012.

[28] S. Kose, S. Tam, S. Pinzon, B. McDermott, B., and E. G. Friedman, E. G., "Active Filter-Based Hybrid On-Chip DC–DC Converter for Point-of-Load Voltage Regulation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 21, No. 4, pp. 680-691, 2013.

[29] M. Salehi, et al. "DRVS: Power-Efficient Reliability Management through Dynamic Redundancy and Voltage Scaling under Variations," *In Proceedings of the international symposium on Low power electronics and design*, 2015, in press.

[30] B. Raghunathan, Y. Turakhia, S. Garg, S., and D. Marculescu "Cherry-picking: exploiting process variations in dark-silicon homogeneous chip multi-processors," *In Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 39-44, 2013.

[31] J. Chinneck, "Practical optimization: a gentle introduction," Electronic document: http://www. sce. carleton. ca/faculty/chinneck/po.html, 2004.