Chapter 3
Variability-Aware Voltage Island Management for Near-Threshold Computing with Performance Guarantees

Ioannis Stamelakos, Sotirios Xydis, Gianluca Palermo, and Cristina Silvano

Abstract The power-wall problem driven by the stagnation of supply voltages in deep-submicron technology nodes, is now the major scaling barrier for moving towards the manycore era. Although the technology scaling enables extreme volumes of computational power, power budget violations will permit only a limited portion to be actually exploited, leading to the so called dark silicon. Near-Threshold voltage Computing (NTC) has emerged as a promising approach to overcome the manycore power-wall, at the expenses of reduced performance values and higher sensitivity to process variations. Given that several application domains operate over specific performance constraints, the performance sustainability is considered a major issue for the wide adoption of NTC. Thus, in this chapter, we investigate how performance guarantees can be ensured when moving towards NTC manycores through variability-aware voltage and frequency allocation schemes. We propose three aggressive NTC voltage tuning and allocation strategies, showing that performance can be efficiently sustained or even optimized at the NTC regime. Finally, we show that NTC highly depends on the underlying workload characteristics, delivering average power gains of 65 % for thread-parallel workloads and up to 90 % for process-parallel workloads, while offering an extensive analysis on the effects of different voltage tuning/allocation strategies and voltage regulator configurations.

Introduction

The end of Dennard's scaling [4] poses designers in front of the so called power/utilization wall. Projections show that the gap between the number of cores integrated on a chip and the number of cores that can be utilized will continue to grow on future technology nodes [8]. As a result, *dark silicon*—transistor count

I. Stamelakos • G. Palermo • C. Silvano
DEIB, Politecnico di Milano, via Ponzio 34/5, Milan 20133, Italy

S. Xydis (✉)
Department of Computer Science, School of Electrical and Computer Engineering,
9 Heroon Polytechneiou, Zographou Campus, 157 80 Athens, Greece
e-mail: sxydis@microlab.ntua.gr

M. Hübner, C. Silvano (eds.), *Near Threshold Computing*,
DOI 10.1007/978-3-319-23389-5_3

under-utilization due to the power budget—has been recently emerged as a major design challenge that jeopardizes the well-established core count scaling path in current and future chip generations.

To address the dark silicon problem, researchers have proposed techniques at the micro-architectural level [10, 11, 27] down to the physical and device level [18, 20]. Near-Threshold voltage Computing (NTC) [6] represents a promising technique to mitigate the effects of dark silicon, allowing a large number of cores to operate simultaneously under a given manycore power envelope. Thus, NTC has emerged as a key enabler for extreme-scale computing platforms [26]. In comparison with the conventional Super-Threshold voltage Computing (STC), computation at NTC regime is performed in a very energy efficient manner, at the expenses of reduced performance and high susceptibility to parametric process variations.

In this chapter, *we investigate the power efficiency potential of manycore architectures at the NTC regime*, *considering process variation as well as power delivery architectures supporting multiple V_{dd} domains*, *under strict performance constraints originated from multicore architectures at the STC regime*. Unlike previous works on variation-aware voltage allocation that target the STC regime [12, 16], we propose the formation of voltage islands (VIs) for the minimization of the impact of within-die variations, which are more evident at NTC, in terms of both performance and power. Then, we show how process variations can be efficiently exploited for further boosting the performance of NTC manycores. To support the aforementioned research objectives, an exploration framework for manycore architectures operating at NTC has been developed to investigate the power efficiency under different workloads, while sustaining the performance when moving from the ST to the NT region.

Evaluation results on both thread-parallel (parallel-application view—high synchronization) and process-parallel (cloud-based application view—low synchronization) workloads show the high dependence of NTC efficiency to the workload's characteristics. Moving to NT regime for a 128-core architecture, while sustaining performance values obtained by a 16-core architecture at STC, average power gains greater than 90 % are delivered for process-parallel workloads, while 65 % power gains for the thread-parallel workload set. We also show that given a best-effort V_{dd} tuning scenario (i.e. let NTC manycore to run faster than the requested STC constraint), a performance improvement of 27 % can be achieved at the expense of 45 % NTC power overhead. However, even with 45 % power overhead, the maximum power dissipated by the NTC manycore is around 10 W. Finally, analyzing the V_{dd} distributions at NTC, we demonstrate that the utilization of multiple VIs together with efficient integrated regulators can be considered a feasible option at NTC to efficiently deal with process variability.

State of the Art

Near-threshold voltage operation relies on the aggressive tuning of the V_{dd} very close to the transistors' threshold voltage V_{th}, to a region where still $V_{dd} > V_{th}$. This decrement of the supply voltage increases the potential for energy efficient computation,

e.g. by reducing V_{dd} from the nominal 1.1 V to 500 mV, energy gains of 10× are reported [6]. NTC is the region that delivers interesting trade offs regarding energy efficiency and transistor delay, since super-threshold V_{dd} quickly reduces energy efficiency while sub-threshold V_{dd} leads to slower transistors. However, NTC comes together with two major drawbacks: (i) reduced performance and (ii) increased sensitivity to process variations.

Performance reduction at NTC is exposed through the limited maximum achievable clock frequency. This is an implicit effect due to the reduction of the $V_{dd} - V_{th}$ difference, applied when moving to the NTC region. Performance degradation can be compensated by exploiting trade-off points corresponding to higher task parallelism at lower clock frequencies. Thus, an important open question to be investigated is the following: *Is the inherent parallelism of applications enough to retain the performance levels of super-threshold design with lower power consumption, thus making it worth going to near-threshold operation*? Pinckey et al. [19] studied the limits of voltage scaling together with task parallelization knobs to address the performance degradation at NTC by considering a clustered micro-architectural template with cores sharing the local cache memory. They proved that under realistic application/architecture/technology features (i.e. parallelization efficiency, inter-core communication, V_{th} selection, etc.) the theoretical energy optimum point $\left(\frac{dEnergy}{dV_{dd}} = 0\right)$ moves from the sub-threshold to the near-threshold region.

Considering a single supply voltage per die, the energy optimum point can be found within an interval of 200 mV higher V_{th}, thus implicitly defining the upper limits of the NTC region.

The second important challenge for manycore architectures operating at NTC regime is their increased sensitivity to process variations. The transistor delay is heavily affected by the variation of V_{th} at NT voltages compared to the one in super-threshold voltages [7, 17]. In addition, failure rate of conventional SRAM cells is increased in low voltage operation [3, 22]. As a consequence, the operating frequency of the cores varies considerably, reducing the yield. In addition, variation's effects on the total power of the chip have to be carefully considered, due to the exponential dependency of leakage current upon V_{th}.

We focus our study on an NTC design space similar to those defined by Dreslinski et al. [6], Karpuzcu et al. [15]. Specifically, we target power efficient NTC manycore architectures that sustain STC performance levels by considering their increased sensitivity to process variation [23]. Performance sustainability is a critical issue for the adoption of the NTC, since best effort approaches are more suitable for managing performance fluctuations due to process variability. In comparison to previous work [6, 15] where only a single system-wide power domain is considered, we differentiate our approach by exploring multiple voltage domain NTC architectures through variation-aware Voltage Island (VI) formation techniques.

Micro-Architecture, Process-Variation and Power Delivery Modelling

Micro-Architecture Model

We focus our study on tile-based architectures, including the ones proposed in Dreslinski et al. [5], Karpuzcu et al. [15] and Stamelakos et al. [25]. Figure 3.1 shows an abstract view of the tile-based manycore architecture, as well as the intra-tile organization. We consider four intra-tile architectures by varying the number of cores per tile and the memory configuration of the last level cache (LLC) per tile. Each core owns a private instruction and data cache (P$). The LLC (LL$) is shared among the different cores composing a tile. The Intel Nehalem processor [13] configuration for the core and the P$ has been adopted. While the P$ size remains constant across the different intra-tile configurations, the size of the (LL$) is scaled according to the number of cores in the tiles, keeping constant the total chip area across the different configurations. We use the following abbreviations for differentiating manycore architectures based on four tile types: (i) S1: each core owns a Last Level LL$, (ii) S2: LL$ is shared between two adjacent cores, (iii) S4: LL$ is shared among four adjacent cores, (iv) S8: LL$ is shared among eight adjacent cores. While S4 and S8 resemble the cluster organizations proposed in Dreslinski et al. [5]

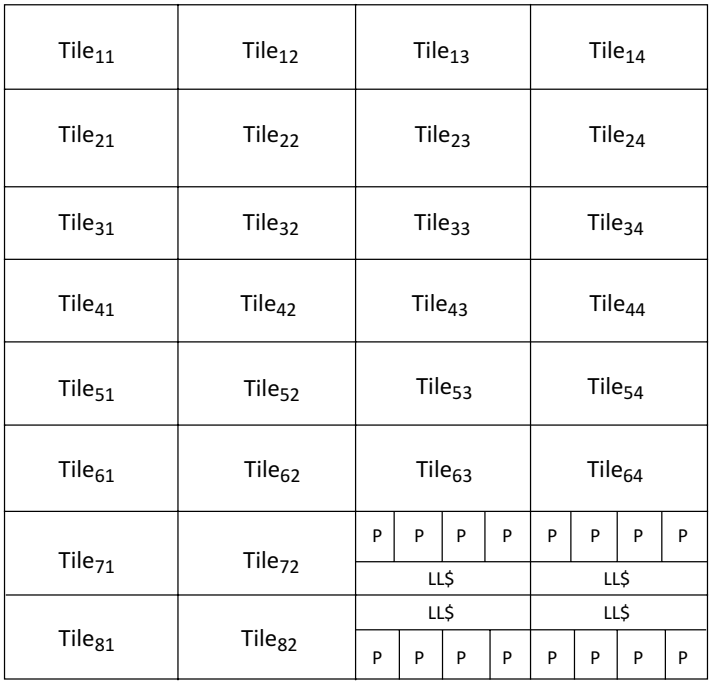

Fig. 3.1 Tile-based manycore architecture

and Karpuzcu et al. [15] we also explored more fine-grained clusters, i.e. S1 and S2. Tile's type defines the minimum VI granularity supported by each manycore configuration.

Process Variation Model

In order to capture the process variation at the NT regime, we integrate the Various-NTV [14] microarchitectural model within the proposed framework. While Various-NTV reuses the spherical distance function in Sarangi et al. [21] for modeling the intra-die spatial correlations, it heavily extends the work done by updating the STC micro-architectural delay and SRAM cell models to reflect in a more accurate manner the higher sensitivity of NTC on process variation. Specifically, (i) it calculates gate-delay following the EKV model [17], (ii) it incorporates a 8T SRAM cell model for reliable read/write operations at NTC and (iii) it considers a larger set of memory timing and stability failure modes. We used ArchFP [9] tool to automatically generate the floorplan of the targeted manycore architectures. Based on the provided manycore floorplan, Various-NTV generates the corresponding variation maps accounting for the within-die (WID) and die-to-die (D2D) process variations. Figure 3.2 shows a sample instance of its V_{th} variation map (Fig. 3.2).

Assuming B as the set of component blocks found in the floorplan and D the set of dies, we now define $V_{th}^{(i,j)}$, $i \in B$, $j \in D$ that corresponds to the V_{th} of the

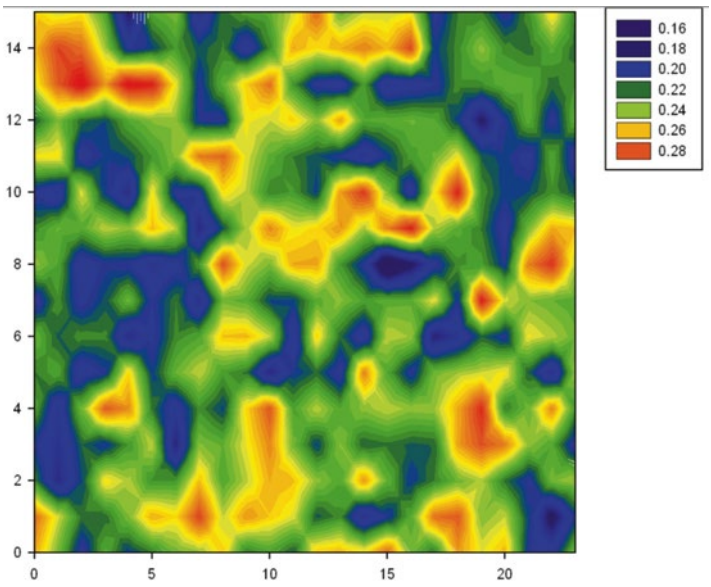

Fig. 3.2 V_{th} variation map corresponding to the tile-based manycore architecture

architecture's component i in the sample die j. Once extracted, $V_{th}^{(i,j)}$ is used for allocating to each component the lowest possible $V_{dd}^{(i,j)}$ for sustaining the f_{NTC} frequency constraint.

Power Delivery Architecture

Generally, the power delivery network can be divided in two components:

1. Off-chip network: one or more power supply rails, powered by off-chip voltage regulators, deliver the appropriate voltages to the chip.
2. On-chip network: a second layer, connected with the off-chip network, consisting of voltage regulators that step down the voltage and deliver it to the cores. The VRs can be of two types:
   - Switching: They have a very good efficiency (~90 %) but they consume a lot of area and they are hard to be integrated on chip.
   - Low Dropout: An LDO is a linear regulator and its efficiency is calculated as follows:

$$\eta_{LDO} = \frac{V_{out}}{V_{in}} \tag{3.1}$$

We consider the power delivery architecture shown in Fig. 3.3. As mentioned in [24], this scheme represents a realistic approach to be used for per-core or per-VI delivery scheme. Initial experimental results reported that the overhead compared with the ideal case where every voltage is precisely delivered would be around 25 % on average. This is because the power supply rail, depending on the platform's variability would have to provide the worst case voltage required leading to a low LDO efficiency. This can be improved by providing extra rails or an extra layer of switching regulators that will downgrade the voltage to an intermediate level. In this case, the experiments show that the overhead will drop to 15 %, which is still quite big but it is a good starting point for improvement and optimization.

Methodology and Framework

Voltage island formation combined with voltage and frequency tuning can provide four different power management schemes, that mitigate variability and deliver different power/complexity trade offs:

1. Single-Voltage/Single-Frequency (SVSF): all the cores have the same voltage and frequency, leading to low complexity implementation but overdesigned power management decisions.
2. Single-Voltage/Multiple-Frequencies (SVMF): the frequency can be tuned individually for each core, enabling in that way the boost or downgrading of the

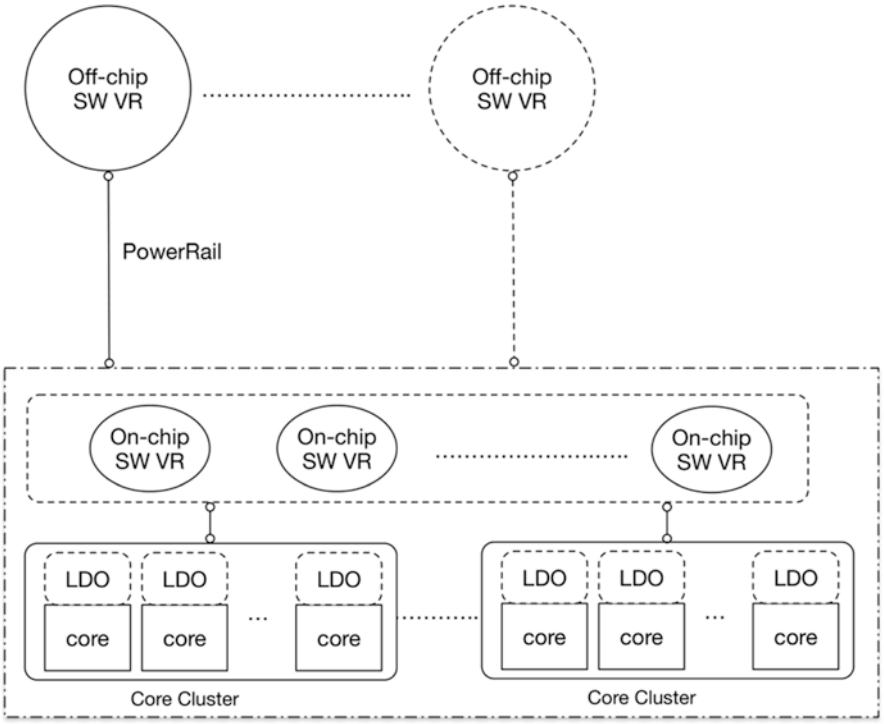

Fig. 3.3 Abstract view of the power delivery architecture

desired cores' performance. However the flexibility of this approach is constrained by the shared voltage.

3. Multiple-Voltages/Single-Frequency (MVSF): voltage scaling can be performed per core or per cluster while the frequency is the same for the whole chip. The benefit of this approach is that the voltage can either be increased so that a higher frequency is achieved or decreased in order to consume less power.
4. Multiple-Voltages/Multiple-Frequencies (MVMF): the two knobs (voltage and frequency) provided in this scheme deliver the benefits of both SVMF and MVSF, leading to big power savings and fine-grained variability reduction, at the expense of high complexity both in implementation and management.

As mentioned before, the effects of process variation are exacerbated in NTC, but except for that, in order to exploit its energy efficiency potential, we should be able to provide performance guarantees to the applications running in an NTC manycore platform, with the ideal case being sustaining their STC performance. This becomes more evident if we consider the emerging paradigms of data center and cloud computing. To further motivate the aforementioned claim, Fig. 3.4 shows the performance distribution for a 128-core NTC many core that implements the best-effort EnergySmart power management SVMF approach [15]. The results are obtained for the executions of the *BARNES* application over 100 different variation

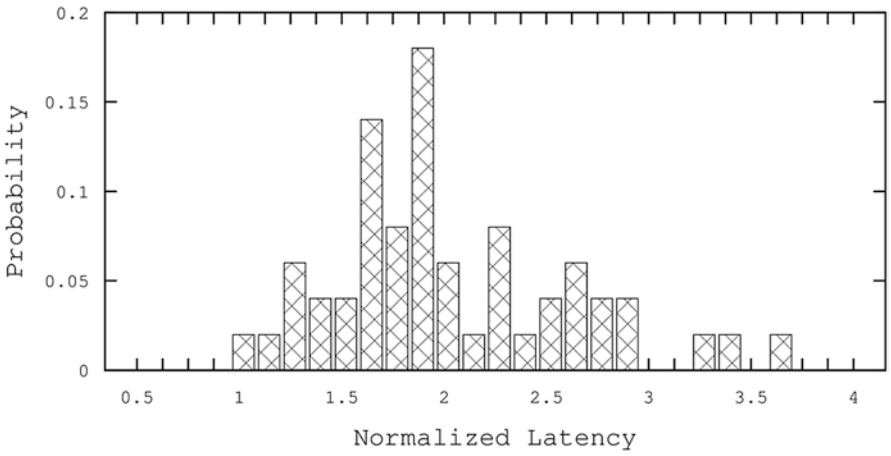

Fig. 3.4 Performance distribution on a 128-core NTC manycore implementing the EnergySmart [15] approach

maps. The normalized performance value of 1 corresponds to the *nominal* performance of the application. As shown, the performance of NTC many core platforms are not controllable and spread out over a wide range of normalized values (from 1 to 3.7) due to the underlying process variability. Thus, the adoption of NTC for applications, exhibiting specific performance and/or throughput constraints, requires careful selection and tuning of the power management scheme. In the following sections, we propose an exploration framework for variation-aware VI formation at NTC and we use it to evaluate several variation-aware power management and tuning strategies that will enable performance sustainability at NTC.

Sustaining STC Performance: Workload Dependent NTC Frequency Assignment

So far, application workloads have been originally developed and characterized for the STC regime. In order to sustain STC performance figures (i.e. latency or throughput) when moving to the NTC regime, the inherent parallelism of the applications should be exploited [19] to alleviate the impact of the reduced clock frequencies at NTC. Assuming a minimum allowed latency L_{min} and maximum core count constraint, C_{max} for the NTC many core, we first calculate the clock frequency of the platform at NTC regime, f_{NTC}, that satisfies the performance constraint. Let $L_{C_{MAX}}$ be the performance, in terms of latency, at the STC regime of a many core architecture with C_{max} number of cores, running at f_{STC}. At STC region, $L_{\min} - L_{C_{MAX}} > 0$ is the available latency slack due to the higher degree of

parallelism of the architecture, that can be exploited to run the application at lower frequency. Utilizing this positive slack, f_{NTC} is calculated as follows:

$$f_{NTC} = \frac{L_{C_{\max}}}{L_{\min}} \times f_{STC} \tag{3.2}$$

The calculated f_{NTC} refers to the target clock frequency of each core at NTC for sustaining STC performance, without considering the spatial effects of process variations. Assuming B as the set of component blocks in the floorplan and D the set of dies, we define $V_{th}^{(i,j)},\ i \in B,\ j \in D$ that corresponds to the V_{th} of the architecture's component i in sample die j. Once extracted, $V_{th}^{(i,j)}$ is used for allocating to each component the lowest possible $V_{dd}^{(i,j)}$ for sustaining the f_{NTC} frequency constraint given that:

$$f_{NTC} \propto \frac{\left(V_{dd}^{(i,j)} - V_{th}^{(i,j)}\right)^{\beta}}{V_{dd}^{(i,j)}} \tag{3.3}$$

where β is a technology-dependent constant (≈1.5). The extraction of the f_{NTC} and the $V_{dd}^{(i,j)}$ per component enables the adoption of different power management schemes for NTC operation with guaranteed performance sustainability

Sustaining STC Performance: VI Formation and Variability Aware V_{dd} Allocation at NTC

Given this NTC scenario, the f_{NTC} and the $V_{dd}^{(i,j)}$ values are used by an MVSF power management scheme to form the voltage island domains and allocate its NTC voltages. The adoption of the MVSF scheme mitigates variability effects, while at the same time it derives an iso-frequency view of the manycore platform. The iso-frequency view of the platform facilitates the application development and porting, because it enables a symmetric platform from the performance point of view. Once the VIs have been defined, we compute the per island V_{dd} assignment that satisfies the f_{NTC} constraint.

More specifically, for the *j*th die, $j \in D$, each VI, $k \in VI$, operates in its own $V_{dd}^{(k,j)}$, tuned for the $VI_{k,j}$ group of processors and memories. $VI_{k,j}$, the core with the highest $V_{th}^{(i,j)},\ i \in B,\ j \in D$ determines the V_{dd} for the specific voltage island, to satisfy the VI_k' s critical path timing. Analyzing the trade off by moving towards coarse grained VI granularities, we reduce area cost since less voltage regulation logic is allocated at the expenses of degrading the power efficiency of the manycore with respect to the finest possible granularity. For $B_k, k \in VI$ the set of resources found in VI_k and from Eq. (3.3), we calculate $V_{dd}^{(k,j)}$ according to the following relation:

$$V_{dd}^{(k,j)} = max_{i \in B_k,\, j \in D}\left[V_{dd}^{(i,j)}\right] \tag{3.4}$$

Exceeding STC Performance: Combining V_{dd} Allocation with Best-Effort f_{NTC} Assignment Under Performance Guarantees

The MVSF approach presented in the previous section guarantees the performance at NTC by allocating in a variability-aware manner the V_{dd} to each VI, in order to enable each VI to run at f_{NTC} (i.e. the minimum clock frequency requested to sustain STC performance without timing violations). However, as shown in Fig. 3.4, the effects of process variability are not monolithic: process variation might generate slower on-chip regions (higher V_{th} values) that reduce the achievable clock frequency as well as faster regions (lower V_{th} values) that enable clock frequencies higher than the f_{NTC} to be allocated. The existence of positive frequency slack at specific regions of the manycore platform can be exploited by moving from the previous MVSF approach to a MVMF power management scheme to further enhance system performance. The adoption of a MVMF scheme enables multiple frequencies to be allocated within a single VI, each one tailored to the performance capabilities of the VI's components, i.e. the underlying tile architecture. However, it is worth noting that MVMF will not impact the V_{dd} allocation of the VIs, which depends on the maximum V_{th} found within each VI, thus performance guarantees continue to be valid. Thus, under the MVMF scenario, the NTC many-core is becoming heterogeneous, by including tiles of processing cores that run at least as fast as f_{NTC} or even faster, implying that the performance is not only sustained, but even optimized with respect to the STC reference configuration.

The frequency allocation within each VI is performed by applying locally the EnergySmart approach [15], since each VI can be considered as an SVMF configuration. Since the $V_{dd}^{(k,j)},\ k \in VI,\ j \in D$, is allocated according to Eq. (3.4), it implies that the maximum achievable frequency, $f_{tile}{}^{(k,j)}$, of each tile within VI_k is bounded as follows:

$$f_{NTC} \leq f_{tile}^{(k,j)} \leq f_{\max}^{(k,j)} \tag{3.5}$$

where $f_{max}{}^{(k,j)}$ corresponds to the maximum frequency supported $V_{dd}{}^{(k,j)}$, and f_{NTC} is the minimum frequency to sustain the performance.

Given the NTC voltage allocation, the power overhead of allowing higher clock frequencies than f_{NTC} is expected to be limited due to the linear but upper bounded frequency increment. We foresee the proposed MVMF scheme to be proved very advantageous for multi-process workloads exhibiting efficient scalability due to limited synchronization, where performance boost of a single core leads to direct throughput improvements.

Experimental Results

In this section, we present the experimental evaluation of the proposed methodology to sustain performance in the near threshold region.

Table 3.1 Experimental setup: platform parameters

Parameter	Value
Process technology	22 nm
STC frequency	3.2GHz
STC supply voltage	1.05 V
Nominal $\frac{V_{th}}{\sigma_{V_{th}}}$.	0.23 V/0.025
Number of cores/core area	128/6 mm^2
Tile/VI size	4 cores/4 tiles
Private cache size/area	320 KB/4.14 mm^2.

Experimental Setup

The Sniper multicore simulator [2] and the McPAT power modeling framework [29] have been used for the performance and power characterization respectively, while the Various-NTV micro-architectural model [14] has been employed to capture the process variation at the NT regime. A summary of the experimental setup used to evaluate the methodology is presented in Table 3.1. Core and caches types, sizes and area are taken from the Intel Nehalem architecture. The target platform is a 128 core chip at NTC (at 22 nm technology node) composed of 32 tiles, each one including four cores and a shared last level cache (LL$) of 8 MB and eight voltage islands (four tiles each). Although in this chapter we are going to present the results obtained by considering single values for the tile size and VI granularity, the approach can be easily generalized to other architectural topologies [25]. Maximum V_{dd} has been set to 1.05 V and the frequency to 3.2 GHz for the STC regime, according to parameter values derived from [1] for conservative technology scaling. By assuming a maximum power budget of 80 W at STC, the performance to be sustained at NTC L_{min} corresponds to a 16 core architecture in the STC regime. From Various-NTV, we extracted 100 different variation maps by using a 24×16 grid based on the core/cache granularity.

Finally, the target applications have been taken from the SPLASH-2 benchmark suite [28], where the "large dataset" workload, provided by Sniper [2], has been adopted. The target applications have been used for the validation in two different scenarios. The first scenario consists of the single application multiple threads (SAMT) approach, where we supposed to run a single application on the platform by using its internal parallelism at thread level (128 threads). The second scenario consists of multiple applications multiple threads (MAMT), where multiple instances of the same application are running (one per tile) and the internal parallelism at the thread-level is used within each tile (four threads). This second version gives a sort of "cloud-oriented" view of the platform. The applications considered in the SAMT version exhibit different behaviors by scaling from 16 to 128 cores: close to ideal *RADIOSITY*, medium *BARNES*, *WATER-NSQ* and limited scaling (*RAYTRACE*, *WATER-SP*). Additionally, we examined an *AVERAGE* case workload, that aggregates in a single execution sequence the five applications, treating them as a

single benchmark. In that way, we manage to see what happens in an average case, where there is a combination of benchmarks that scale well and others that don't scale well. On the opposite, all the applications in the MAMT version present an almost ideal scaling passing from 16 cores (2 application instances over 2 tiles) to 128 cores (32 application instances).

Power Estimation for the NTV Regime

Given the V_{dd} allocation per VI from Eq. (3.4), $V_{dd}^{(k,j)},\ k \in VI,\ j \in D,$ and the power characterization for the many-core with C_{max} number of cores at STC, we can calculate the power of each component in NTC. For $i \in B,\ j \in D,\ k \in VI,$ the dynamic, *DP* and leakage, *LP*, power scaling factors are:

$$SF_{DP}^{(i,,j,,k)} = \left(\frac{V_{dd}^{(k,j)}}{V_{dd_{STC}}}\right)^2 \times \left(\frac{f_{NTC}}{f_{STC}}\right) \tag{3.6}$$

$$SF_{LP}^{(i,,j,,k)} = \left(\frac{V_{dd}^{(k,j)}}{V_{dd_{STC}}}\right) \times \exp\left(\frac{Vth_{STC} - V_{th}^{(i,j)} + DIBL}{n \times V_{thermal}}\right) \tag{3.7}$$

$$DIBL = \lambda\left(V_{dd}^{(k,j)} - V_{dd_{STC}}\right) \tag{3.8}$$

where *DIBL* is the coefficient modeling the Drain-Induced Barrier Lowering effect, $V_{thermal}$ is the thermal voltage, and n is the sub-threshold slope coefficient. The DIBL effect is a deep-submicron effect related to the reduction of the threshold voltage as a function of the drain voltage. DIBL is enhanced at higher drain voltage and tends to become more severe with process scaling to shorter gate lengths. Lowering supply voltage provides an exponential reduction in sub-threshold current resulting from the DIBL effect. Figure 3.5 shows the impact of DIBL effect on the

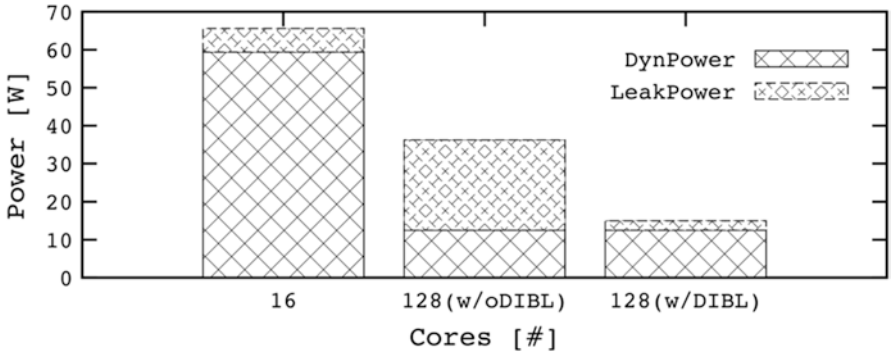

Fig. 3.5 Power breakdown for STC-16core and NTC-128core architectures with and without DIBL effect

reduction of leakage power in many core architectures at NTC regime. As shown, by moving from STC multi-core (16 cores) to NTC manycore (128 cores) architecture configurations, the DIBL effect accounts for a significant portion of the total power of the system.

Power Gains: NTC vs STC

Figure 3.6 shows the power consumption comparison when passing from 16 cores at STC to 128 cores at NTC for each benchmark in both SAMT and MAMT versions. The power values for the same benchmark on SAMT and MAMT versions are not comparable because of application performance are different in the two cases. All the MAMT versions of the applications and the *RADIOSITY-SAMT* deliver large power gains (>90 %) due to the almost ideal performance scaling as the number of cores increases. The rest of the applications in SAMT version present a power gain that depends on the scaling capability, since it impacts the minimum frequency to be sustained and thus the minimum V_{dd} to be deployed to the voltage islands. For the remaining applications, Fig. 3.6 shows a 75 % decrement in power for *BARNES* and *WATER-NSQ*, around 25 % for *WATER-SP* and an almost identical power for *RAYTRACE*. The *AVERAGE-SAMT* workload (composed of a sequential mix of all applications) delivers a power gain of 65 %.

Variation Aware Versus Overdesign NTC Operation

We compared the power gains delivered by the proposed variation aware VI formation versus an overdesign approach to mitigate variation effects. From the V_{th} distribution, we calculate the V_{dd} of architectural components according to Eq. (3.3), with V_{th}'s

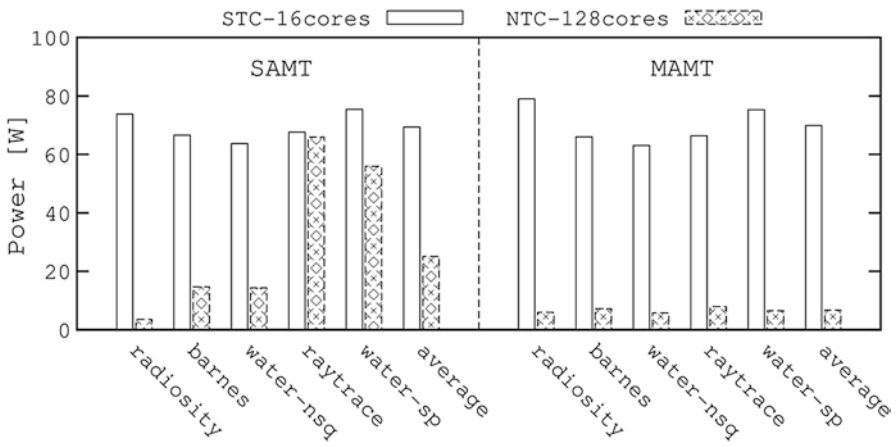

Fig. 3.6 Power consumption: 16-core STC chip versus 128-core NTC for both SAMT and MAMT versions of the target applications

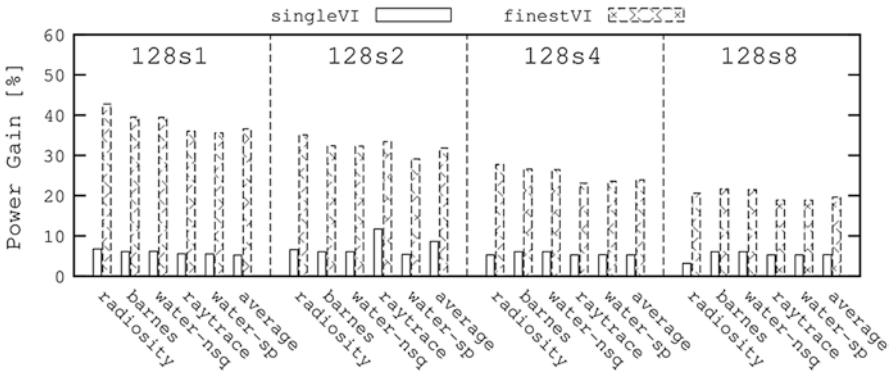

Fig. 3.7 Power gains of variability-aware NTC technique w.r.t. overdesign

overdesign value being equal to $\mu_{V_{th}} + 3\sigma_{V_{th}}$. Figure 3.7 reports the gains of the variability aware approach over the overdesign one. The histograms with the *singleVI* annotation represent power gains when having only one VI, and as a consequence one V_{dd} for the whole chip. Under a *singleVI* configuration, the variation aware approach achieves power gains around 5 %, for all the available cluster architectures $S_i, i \in \{1,2,4,8\}$. On the contrary, the histograms with the *finestVI* annotation show the power gains achieved by considering the finest VI granularity possible for each architecture. Since S1 enables the finest 1×1 VI granularity to be exploited, it delivers the highest gains over the overdesign approach, that range between 34 and 42 %. In the rest of the architectures, namely (S2, S4, S8), the gains vary between 29–34 %, 25–28 % and 18–23 %, respectively.

Relaxing the Isofrequency Constraint

Figures 3.8 and 3.9 show the power/performance impact of the relaxation on the isofrequency constraint. To better evaluate this scenario, we present the experimental data considering only the MAMT version of the *average* case. As stated in the previous section, while the MVMF has ideally an advantage due to the increment of the tile frequency, this can be really exploited only when the application is aware of this performance asymmetry. This is not the case of the SAMT version of our target applications. To get a clear view of the performance improvement we adopted the application throughput concept as the rate of jobs (application instances) completed within a time interval. As expected, the MVMF approach offers a performance speedup due to the frequency increment in the tiles not affected by the critical V_{th}. However, the performance improvement (~27 %) is balanced by an increased power

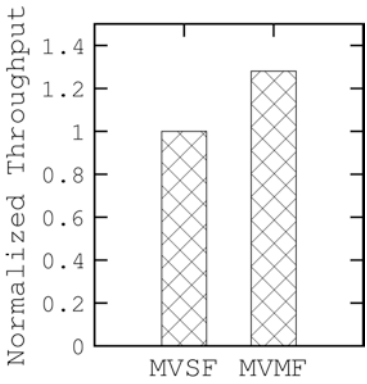

Fig. 3.8 Impact of MVMF vs MVSF in terms of throughput

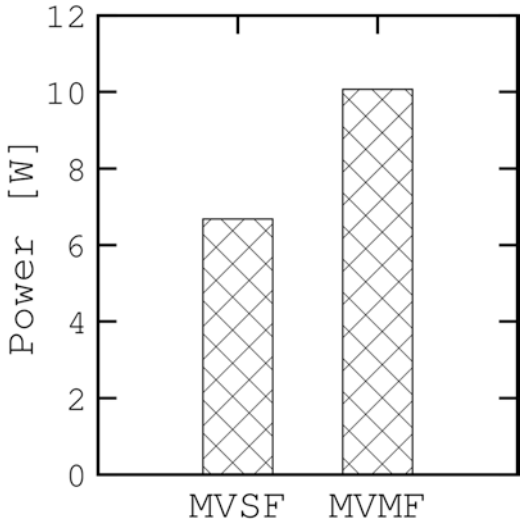

Fig. 3.9 Impact of MVMF vs MVSF in terms of power

overhead (~45 %). The larger power improvement than the performance advantage, is due to the resource sharing among the tiles after the LL$ that limits the throughput.

Additionally, Fig. 3.10 shows the tile frequency distribution across the 100 variation maps by using the MVMF mode. The minimum frequency is 400 MHz to guarantee the application performance in terms of throughput. As expected, the minimum value is the most probable because there is at least one tile per VI (the one that limits the V_{dd} scaling) running at that frequency. Regarding the other values, we can notice that the distribution shows a long tail meaning that there is a large margin that can be used for further speedups.

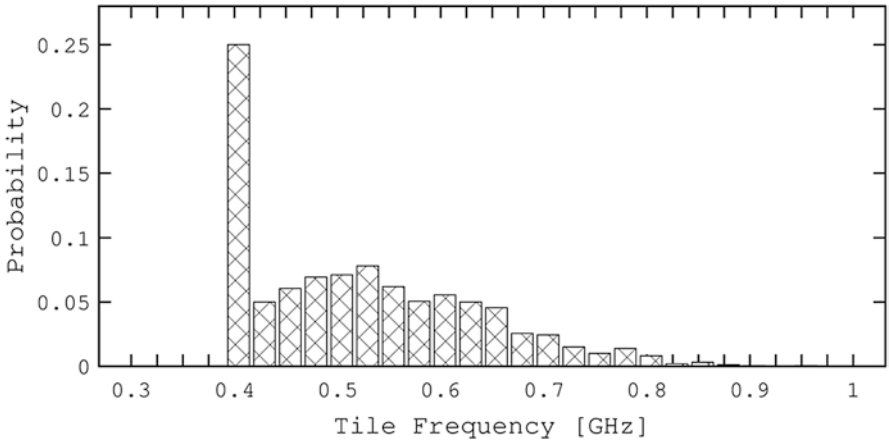

Fig. 3.10 Tile frequency distribution in MVMF mode

Impact of Power Delivery Architecture

The analysis conducted so far considers an ideal scenario where all the requested on-chip voltage levels can be delivered precisely. As a first step we analyzed three different voltage regulator resolutions, delivering voltage with a precision of (i) 12.5 mV, (ii) 25 mV and (iii) 50 mV. Figure 3.11 respectively presents: the average power overhead for each voltage regulator precision in Fig. 3.11a and the V_{dd} distribution according to each regulator resolution in Fig. 3.11b–d. The power overhead and the V_{dd} distributions have been calculated across the 100 variation maps considering a target frequency of 400 MHz to be sustained.

In Fig. 3.11a we refer to power overhead as the normalized average difference between the power consumed in the ideal case (voltage regulator delivering arbitrary V_{dd} values) and the power corresponding to specific values of voltage precision. As expected, the higher the resolution the smaller the overhead since we are closer to the ideal case, passing from a 12 % at 50 mV to less than 3 % at 12.5 mV. This limited overhead value is interesting also considering the results shown in Fig. 3.11b–d, where it can be noticed that the V_{dd} distribution is very concentrated, suggesting that the voltages can be distributed by a few power rails and/or voltage regulators.

Nevertheless, this work's goal was not to design or suggest a power delivery architecture, but in this section our intention was to demonstrate that there are feasible solutions that can be further explored in order to obtain a power efficient manycore platform.

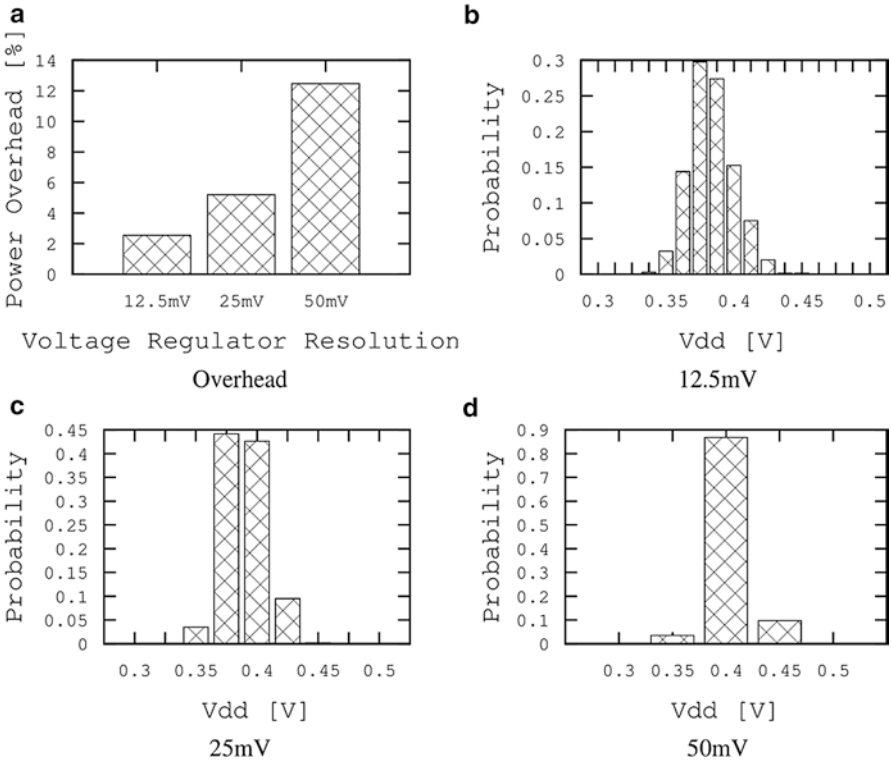

Fig. 3.11 Voltage regulator analysis: Power overhead (**a**) and V_{dd} probability distribution (**b**–**d**) for three voltage regulator resolutions

Conclusion

This chapter focuses on the emerging NTC paradigm as a key enabler for the power-efficient scaling of many core architectures. While power efficiency is guaranteed by definition at the NTC regime, performance guarantee is still an open challenge. Sustaining STC performance figures during NTC operation is a critical issue for the wider adoption of the NTC paradigm. Towards this direction, we presented a set of techniques for variability-aware voltage island formation and voltage/frequency tuning that enable moving to NTC regime while sustaining STC performance guarantees. Extensive experimentation showed the optimization potentials of moving towards near-threshold voltage computing, outlining its high dependency on both workload characteristics and voltage tuning strategy.

References

1. Borkar S (2010) The exascale challenge. In: 2010 International Symposium on VLSI design automation and test (VLSI-DAT), pp 2–3
2. Carlson TE, Heirman W, Eeckhout L (2011) Sniper: exploring the level of abstraction for scalable and accurate parallel multi-core simulations. In: International conference for high performance computing, networking, storage and analysis (SC)
3. Chang L, Montoye R, Nakamura Y, Batson K, Eickemeyer R, Dennard R, Haensch W, Jamsek D (2008) An 8T-SRAM for variability tolerance and low-voltage operation in high-performance caches. IEEE J Solid State Circuits 43(4):956–963
4. Dennard R, Gaensslen F, Rideout V, Bassous E, LeBlanc A (1974) Design of ion-implanted MOSFET's with very small physical dimensions. IEEE J Solid State Circuits 9(5):256–268
5. Dreslinski RG, Zhai B, Mudge TN, Blaauw D, Sylvester D (2007) An energy efficient parallel architecture using near threshold operation. In: PACT, pp 175–188
6. Dreslinski RG, Wieckowski M, Blaauw D, Sylvester D, Mudge TN (2010) Near-threshold computing: reclaiming Moore's law through energy efficient integrated circuits. Proc IEEE 98(2):253–266
7. Eisele M, Berthold J, Schmitt-Landsiedel D, Mahnkopf R (1997) The impact of intra-die device parameter variations on path delays and on the design for yield of low voltage digital circuits. IEEE Trans Very Large Scale Integr Syst 5(4):360–368
8. Esmaeilzadeh H, Blem E, St Amant R, Sankaralingam K, Burger D (2011) Dark silicon and the end of multicore scaling. In: Proceedings of the 38th annual international symposium on computer architecture, ISCA'11, pp 365–376
9. Faust GG, Zhang R, Skadron K, Stan MR, Meyer BH (2012) ArchFP: rapid proto-typing of pre-RTL floorplans. In: Katkoori S, Guthaus MR, Coskun AK, Burg A, Reis R (eds) VLSI-SoC, pp 183–188
10. Goulding-Hotta N, Sampson J, Venkatesh G, Garcia S, Auricchio J, Huang P, Arora M, Nath S, Bhatt V, Babb J, Swanson S, Taylor M (2011) The GreenDroid mobile application processor: an architecture for silicon's dark future. IEEE Micro 31(2):86–95
11. Govindaraju V, Ho CH, Sankaralingam K (2011) Dynamically specialized datapaths for energy efficient computing. In: 2011 IEEE 17th international symposium on high performance computer architecture (HPCA), pp 503–514
12. Herbert S, Garg S, Marculescu D (2012) Exploiting process variability in voltage/frequency control. IEEE Trans Very Large Scale Integr Syst 20(8):1392–1404
13. Kanter D (2008) Inside Nehalem: Intel's future processor and system. http://www.realworldtech.com
14. Karpuzcu UR, Kolluru KB, Kim NS, Torrellas J (2012) VARIUS-NTV: a microarchitectural model to capture the increased sensitivity of manycores to process variations at near-threshold voltages. In: IEEE/IFIP international conference on dependable systems and networks, DSN, pp 1–11
15. Karpuzcu UR, Sinkar AA, Kim NS, Torrellas J (2013) EnergySmart: toward energy-efficient manycores for near-threshold computing. In: HPCA, pp 542–553
16. Majzoub SS, Saleh RA, Wilton SJE, Ward RK (2010) Energy optimization for many-core platforms: communication and PVT aware voltage-island formation and voltage selection algorithm. Trans Comput Aided Des Integr Circuits Syst 29(5):816–829
17. Markovic D, Wang C, Alarcon L, Liu TT, Rabaey J (2010) Ultralow-power design in near-threshold region. Proc IEEE 98(2):237–252
18. Paterna F, Reda S (2013) Mitigating dark-silicon problems using superlattice-based thermoelectric coolers. In: Proceedings of the conference on design, automation and test in Europe, EDA Consortium, San Jose, CA, USA, DATE'13, pp 1391–1394
19. Pinckney N, Sewell K, Dreslinski RG, Fick D, Mudge T, Sylvester D, Blaauw D (2012) Assessing the performance limits of parallelized near-threshold computing. In: Proceedings of the 49th design automation conference, pp 1147–1152

20. Raghavan A, Luo Y, Chandawalla A, Papaefthymiou MC, Pipe KP, Wenisch TF, Martin MMK (2012) Computational sprinting. In: IEEE HPCA, pp 249–260
21. Sarangi S, Greskamp B, Teodorescu R, Nakano J, Tiwari A, Torrellas J (2008) VARIUS: a model of process variation and resulting timing errors for microarchitects. IEEE Trans Semicond Manuf 21(1):3–13
22. Sasan A, Homayoun H, Eltawil AM, Kurdahi FJ (2011) Inquisitive defect cache: a means of combating manufacturing induced process variation. IEEE Trans Very Large Scale Integr Syst 19(9):1597–1609
23. Silvano C, Palermo G, Xydis S, Stamelakos IS (2014) Voltage island management in near threshold manycore architectures to mitigate dark silicon. In: Design, automation & test in Europe conference & exhibition, DATE 2014, Dresden, Germany, March 24–28, 2014, pp 1–6
24. Sinkar AA, Ghasemi HR, Schulte MJ, Karpuzcu UR, Kim NS (2014) Low-cost per-core voltage domain support for power-constrained high-performance processors. IEEE Trans Very Large Scale Integr Syst 22(4):747–758
25. Stamelakos I, Xydis S, Palermo G, Silvano C (2014) Variation aware voltage island formation for power efficient near-threshold manycore architectures. In: Proceedings of the ASP-DAC, ASP-DAC'14
26. Torrellas J (2014) Extreme-scale computer architecture: energy efficiency from the ground up. In: Proceedings of the conference on design, automation and Test in Europe, DATE'14
27. Turakhia Y, Raghunathan B, Garg S, Marculescu D (2013) HaDeS: architectural synthesis for heterogeneous dark silicon chip multi-processors. In: DAC, ACM, pp 173–178
28. Woo SC, Ohara M, Torrie E, Singh JP, Gupta A (1995) The SPLASH-2 programs: characterization and methodological considerations. SIGARCH Comput Arch News 23(2):24–36
29. Li S, Ahn JH, Strong RD, Brockman JB, Tullsen DM, Jouppi NP (2009) McPAT: an integrated power, area, and timing modeling framework for multi-core and many-core architectures. In: Proceedings of the 42nd annual IEEE/ACM international symposium on Microarch20tecture, MICRO 42, pp 469–480