Thermal-Aware Task Scheduling for Peak Temperature Minimization under Periodic Constraint for 3D-MPSoCs

Vivek Chaturvedi^{*}, Amit Kumar Singh[†], Wei Zhang[‡] and Thambipillai Srikanthan^{*} Nanyang Technological University, Singapore^{*}, National University of Singapore, Singapore[†] Hong Kong University of Science and Technology, Hong Kong[‡] Email: *{vchaturvedi, astsrikan}@ntu.edu.sg, [†]amit.singh@nus.edu.sg, [‡]wei.zhang@ust.hk

Abstract—3D-MPSoC offer great performance and scalability benefits. However, due to strong vertical thermal correlation and increased power density, thermal challenges in 3D-MPSoC are critical. In this paper, we propose a novel thermal aware task scheduling technique that combine intelligent task mapping with DVFS to minimize the peak temperature of the system. Particularly, our approach leverages on the fundamental thermal characteristics of 3D architecture when mapping tasks to processing cores and employing DVFS at design time followed by a simple thermal optimization step at run time. Our experiments validate the efficiency of our approach in peak temperature minimization up to $14^{\circ}C$ compared to other existing methods.

I. INTRODUCTION

Three-dimensional (3D) integrated circuits (ICs) have attracted attention of several researchers due to their numerous advantages. In a 3D IC, multiple layers of logic are stacked vertically and are generally connected using Through Silicon Vias (TSVs) [7]. Such stacking significantly reduces the die size and average interconnect wire lengths. This in turn reduces production costs, communication delays and interconnect energy, while increasing the interconnect flexibility in comparison to traditional 2D counterparts [12]. Therefore, 3D ICs alleviate performance bottleneck problems incurred due to on-chip interconnects that do not scale in proportion to the process technology [18]. To summarize, 3D ICs provide attractive alternatives to implement multi-core based systems like MPSoCs.

However, despite the significant advantages, 3D ICs suffer from serious thermal problems due to the increased power density as a result of vertical stacking of active layers on top of each other. The thermal problem, *i.e.*, high temperature, will increase leakage power and decrease system performance, reliability and lifetime. Therefore, it is of paramount importance to minimize the thermal problems throughout the design process. Researchers have paid extensive attention to mitigate thermal problems by considering several aspects such as liquid cooling [8], efficient thermal via placement [14], dynamic thermal management (DTM) [3], thermal-aware task allocation and scheduling [15, 17], etc. However, the dynamic mechanisms, such as DTM or dynamic task allocation, lead to unpredictable behavior for the software execution, which might result in missed deadlines. Therefore, for timeconstrained applications widely used in embedded systems, it is important to take the deadline constraint into consideration in the thermal-ware mapping process.

Existing approaches to perform thermal-aware mapping of applications employ various task allocation and scheduling

techniques. A significant attention among them has been paid to 2D multi-cores [10]. However, a straightforward extension of such approaches to 3D multi-cores will not lead to good mapping results due to the significantly different thermal characteristics of 3D ICs. Most of the prior proposals for thermal management on 3D multi-cores focuses on independent tasks, including the thermal-ware task allocation, Dynamic Voltage/Frequency Scaling (DVFS), clock gating and hot task migration [3, 17, 19, 22]. Recently, there has been focus on the connected tasks described as dataflow graphs, e.g., Directed Acyclic Graphs (DAGs) [5, 11, 15]. In these works, optimizations are performed considering both thermal issues and communication overhead. However, most of these works do not consider delay or throughput constraints. The few ones which take the throughput constraint into account employ scheduling techniques ignoring task characteristics, such as in [15] and [5]. Finally, the leakage power which closely relates to the thermal issues, has been missing from the power models of all the above mentioned works.

In order to address the shortcoming of the existing thermalaware mapping approaches, we propose a two-stage thermalaware mapping algorithm for throughput-constraint applications on 3D multi-cores with both design time optimization and run-time adjustment. Based on a complete power model with leakage power effects, in the static optimization stage, we first assign dataflow graphs to suitable layers to achieve power balance and meet the throughput constraints. Then available slacks for the tasks are exploited to perform DVFS for further peak temperature reduction. Following the static stage, a runtime thermal optimization strategy is applied using run-time information to exchange the workload from hot core stacks to cool stacks without affecting the periodic constraint of the application. Our contributions can be summarized as follows:

- A thermal-aware mapping approach to map periodic constrained applications described as DAG.
- Consideration of leakage power effects on the temperature profile.
- 3) A DVFS strategy applied to the mapping results based on the available slack between execution length and periodic constraint.
- 4) An on-line optimization technique to further reduce the peak temperature.

To the best of our knowledge, this is the first work that addresses thermal-aware mapping on periodic constrained applications on 3D multi-cores while considering DVFS enabled cores and leakage power effects, and performing both offline and on-line optimization. The experimental results demonstrate significant peak temperature reduction compared to other existing approaches.

Organization: In section II, related works regarding the mapping of applications on 3D multi-cores are discussed. The architecture, application and power models are introduced in section III. The proposed thermal-aware mapping flow is discussed in section IV. Experimental results are presented in section V. Finally, Section VI concludes the paper.

II. RELATED WORK

Thermal optimization of 3D ICs has become attractive to several researchers in order to overcome the issues caused due to thermal emergencies, for example, threatening to reliability and performance. In addition to focusing on thermal-aware 3D IC designs [14, 23], researchers have focused on thermalaware mapping and scheduling as well. The approaches reported in the literature can be categorized into static (designtime) and dynamic (run-time) techniques.

Static techniques aim to find a thermally balanced mapping at design time and normally use a model of the physical chip or thermal behavior of 3D ICs. Addo-Quaye et al. [2] propose a genetic algorithm to find thermally balanced mappings while taking temperature and communication load into account. In the similar direction, Cheng et al. [11] use a combination of heuristics, simulated annealing and a greedy algorithm to find optimal static mappings, and show that a trade-off exists between interconnect energy consumption and peak temperature. However, in these approaches, application constraints such as throughput requirement have not been considered. Sun et al. [22] consider applications with deadlines and propose a technique to find optimal mappings, where an initial mapping is first found by employing a power balancing algorithm and then iterative improvement is performed by simulating the temperature distribution and migrating tasks. However, communication between tasks is not taken into account.

Dynamic techniques take actions to minimize hotspots and thermal gradients (spatial and/or temporal) based on the current temperature distribution in the chip. Zhu et al. [27] exploit workload power characteristics and processor core thermal characteristics for efficient thermal management. Coskun et al. [3] propose a run-time task assignment algorithm that takes the thermal history of cores into consideration. In [19], concept of thermal herding has been used, where hot jobs are assigned to the cores close to the heat sink. A thermal-aware OSlevel scheduler for 3D multi-cores is proposed in [26]. These techniques however do not consider deadline constraints and the effect of inter-task communication.

III. PRELIMINARIES

In this section, we provide the details of our system models including application model, architecture model of 3D multiprocessor, power and thermal models. We also define the terms which will be used throughout the paper.

A. Application Model

The applications evaluated in this work are modeled as DAG, which are very efficient in representing both embedded and general purpose applications [6]. An application G(V, E)

contains a set of vertexes V denoting the task (node) to be executed, and edges E denoting the precedence relationship between tasks. Each vertex is associated with its worst-case execution cycles and each edge with worst-case communication cycles, which are assumed to be known at design time. Applications are assumed to be periodic in nature and their performance is guaranteed by a periodic constraint (L). An advantage of DAG model is that multiple applications can be easily combined as a single DAG application [24]. As a general model, we also simplify our application model as a single DAG application with a periodic (throughput) constraint L.

B. Power Model

We assume each core is separately DVFS enabled and able to run in different modes, with each mode being characterized by a pair of parameters (v_{k_i}, f_{k_i}) . Here v_{k_i} is the supply voltage and f_{k_i} is the working frequency in mode k_i , and the overall power consumption (in *Watt*) is composed of the dynamic power P_{dyn} and leakage power P_{leak} . In our power model, P_{dyn} is independent of the temperature, while P_{leak} is sensitive to the temperature. The dynamic power consumption of core *i*, \mathcal{P}_i can be formulated as in [21]

$$P_{dyn,i} = \gamma_{k_i} \cdot v_{k_i}^2 \cdot f_{k_i} \tag{1}$$

where v_{k_i} and f_{k_i} is the supply voltage and frequency of processing core C_i and γ_{k_i} is a constant. Similar to the work in [20], we model the leakage power of processor \mathcal{P}_i as follows:

$$P_{leak,i} = \left(\alpha_{k_i} + \beta_{k_i} \cdot T_i(t)\right) \tag{2}$$

where α_{k_i} and β_{k_i} are constants depending on the processor running mode, *i.e.*, mode k_i .

C. Architecture Model of 3D Multi-Processor

We model our 3D-MPSoC as a regular mesh of tiles stacked over multiple layers connected by Network-on-Chip (NoC) as depicted in Figure 1. Every tile contains a processing core connected to a network router. Further, each core block is divided into computation blocks and L2 cache. Lateral communication is carried out via NoC, while vertical links are implemented using through-silicon-vias (TSV). There could be several ways to select the type and number of TSVs across layers and different ways of implementation for vertical links, however in this work we followed the implementation style demonstrated in [5] and [4].

In order to perform thermal simulations for a given application, we supply floorplan and power distribution details of this model to HotSpot [1], a widely used thermal simulation tool, and extract on-chip thermal profile. The architecture file to the simulator provides all the required information on physical properties of 3D multi-core platforms including the heat sink and heat spreader. The details of all the thermal parameters are listed in Section V.

D. Definition of Terms

- 1) *Hot Node*: It is defined as the task node with computation demand (*i.e.*, execution cycles) higher than the average demand of the overall application G(V, E). Such a node will contribute more towards peak temperature.
- 2) *Periodic Ratio*: It is defined as the ratio of the periodic constraint L to the length of the critical path of the application G(V, E).



Fig. 1. Two-layer 3D NoC architecture with 3D-mesh topology

IV. THERMAL-AWARE TASK MAPPING ALGORITHM

In this section, we introduce our proposed off-line/on-line thermal aware task mapping algorithm for peak temperature minimization. Ours is a two-stage approach that combines design time mapping and DVFS strategy with run-time thermal optimization methodology. Before presenting our thermalefficient mapping algorithm, we first present a thermal profiling approach to exploit the architecture characteristics that enables wiser decision making when mapping tasks to cores on 3D platforms.

A. Thermal Profiling

As mentioned previously, 3D architectures have strong vertical thermal correlation that translates to severe thermal challenges compared to 2D architectures. When solving thermal issues in 2D, power balancing across the platform is widely applied to provide acceptable solutions. Unfortunately, power balancing in 3D architecture becomes tricky as the impact of the same power consumption on two similar cores located on two different vertical layers result in drastically different thermal profiles. Moreover, previous works have shown that the heat dissipation from different layers depends upon their proximity to the heat sink [5, 9]. These observations promote that the power consumption at each layer should be made different depending upon its proximity to the heat sink. Therefore, an investigation on the power distribution pattern on a 3D platform is relevant to facilitate intelligent task mapping for thermal efficiency. Hence, we introduce a thermal profiling methodology to collect the power distribution pattern that enables the most suitable core selection for task mapping to maintain peak temperature within the thermal constraint.

In this approach, we first run all the cores on the highest mode of execution consuming maximum power. We collect the steady state temperature for this power distribution. Thereafter, we run a self corrective algorithm that updates the power trace after every steady state temperature simulation by decreasing the power of the cores with the temperature higher than the average temperature of the chip, and increasing the power of the cores with the temperature lower than the average temperature of the chip. The algorithm continues to scale the power on the cores until the peak temperature of the chip is less than the thermal constraint and the thermal gradient is within acceptable value. The details of this approach is

ALGORITHM 1: Thermal Profiling Algorithm

Input: 3D MPSoC model, stopping criterion $\delta \in \mathbb{R}$, thermal constraint $T_{const} \in \mathbb{R}$, dynamic power levels (DVFS based) $P \in [0, P_{max}] P \in \mathbb{R}$ **Output:** Target power levels PL_t , $t \in [0, N_{tiles} - 1]$. Initialize power levels $\forall t \in [0, N_{tiles} - 1] : Pl_t = P_{max};$ $T_{avg} = 0, T_{min} = 0, T_{max} = 1000;$ i = 0: while $(T_{max} \ge T_{const})$ and $(T_{max} - T_{min}) \ge \delta$ do | Generate power traces for all blocks based on Pl_t ; Simulate steady state temperature dist.; $T_{avg} \leftarrow$ average chip temperature; $\begin{array}{l} T_{avg} \leftarrow \text{maxe composition}, \\ \forall t \in [0, N_{tiles} - 1] : T_{peak,t} \leftarrow \text{max. temp. in tile } t; \\ T_{max} \leftarrow \max(T_{peak}); \end{array}$ $T_{min} \leftarrow \min(T_{peak})$ if $(T_{max} \leq T_{const})$ then break end for all tiles $t \in [0, N_{tiles} - 1]$ do if $(T_{peak,t} \leq T_{avg})$ then $|Pl_t = Pl_t + +$ Increase power level (DVFS mode) end if $(T_{peak,t} \ge T_{avg})$ then $|Pl_t = Pl_t - - \text{Decre}$ - Decrease power level (DVFS mode) end end i +end **return** power levels P_L

provided in Algorithm 1. This algorithm is an extension of the work presented in [5]. In our work, we modified the approach to fit our assumption of discrete voltage levels available for each core. Next, we discuss our proposed offline/online algorithm in detail.

B. Off-line Optimization

Assuming an application G(V, E), for instance as shown in Figure 2(A), available with its timing parameters known at design time, our off-line algorithm follows three steps to achieve a thermal-aware task mapping with DVFS level assigned to each task. Next, we discuss each step in detail with the help of a representative example. In this example, we assume a very simple 3D platform with one core each on two layers stacked together and the periodic constraint L for the application is equal to 21 units.

- 1) Priority Assignment: Given an application G(V, E), we first create a task execution order based on the static priority level. The static level of a task node is defined as the maximum path length from the node to the sink node. When calculating the path length, we ignored weight on edges of the path. This step is a widely used approach to assign priorities to DAGs, *i.e.*, critical level of the nodes, for multi-processor scheduling [13]. As shown in 2(B), we calculate the static level for each task and prioritize the node with the highest static level. If two nodes have same static level then we prioritize the one with higher worst case execution cycles.
- 2) *Thermal-Aware Task Mapping*: Once the priority list of the application is ready, we start mapping each task on the available processors. To do this, we follow a set of rules:
 - (*i*): If node is a *Hot node* then it should be mapped on the layer closest to heat sink and to the core with earliest start time to allow longer execution

with smaller heat accumulation, under the condition of no periodic constraint violation caused. However, in the possibility of constraint violation, assign the task to the next layer closer to heat sink, and so on until the constraint is met. The rationale behind this approach is to avoid excessive power consumption at layers away from the heat sink, as well as satisfy the constraint.

• (*ii*): If node is a regular task then it should be mapped to the core with earliest starting time in the system, since their contribution to the heat generation is relatively small. In case of multiple available cores, load distribution based on the results of thermal profiling will be done. Since thermal profiling has given the power distribution among the cores to meet the temperature constraint, the task can be assigned to the core with more available power budget. The goal of the above steps is to maximize the slack time between periodic constraint and critical path to allow possible DVFS in later step.

The mapping of tasks in our example according to the above rules is shown in Figure 3(A).

3) Slack Distribution and DVFS Assignment: In order to achieve desirable performance, each application is bound to maintain the periodic constraint L at all times. While performing thermal-aware task mapping, we make sure that in no circumstances this constraint is violated. At the same time, we also create an opportunity to claim slack time between the critical execution path of the application and the periodic constraint. The slack time achieved can be efficiently used to apply DVFS at each core to further improve the thermal behaviour of the system.

We begin this step by identifying the critical path on the mapped application. We calculate the $path_{slack}(i)$ factor for each path *i* starting from the entry node to the leaf node as follows:

$$path_{slack}(i) = L + t_{start(i)} - t_{end(i)}$$
(3)

where $t_{start(i)}$ and $t_{end(i)}$ are the start time and end time of the current path and L is the periodic constraint. The critical path is the path with minimum $path_{slack}$.

Once the critical path is identified, we next calculate the task scaling factor, $Task_{Scale}$, for the critical path and multiply it with all the task nodes of the application. By doing this we distribute the available slack time among each task relative to their computation requirement. Here $Task_{Scale}$ is given by:

$$Task_{scale} = \frac{L + t_{start} - t_{end}}{\sum_{tasksonpath} Task_{execution}}$$
(4)

If there are multiple critical paths, the one with smaller $Task_{Scale}$ will be used. $Task_{Scale}$ can be seen as a speed scaling factor. Assuming all tasks are running on maximum speed, then when slack is distributed to each task, the speed of execution can be slowed down by this factor without violating periodic constraint. Hence, based on $Task_{Scale}$, we select the DVFS level for each



Fig. 2. Priority Assignment



Fig. 3. Task Mapping and DVFS Assignment

task of the application such that the periodic constraint is not violated and schedule the application for execution. As shown in Figure 3(B), the originally mapped tasks of 3(A) are stretched using Equations 3 and 4.

C. On-Line Algorithm

The impact of large thermal gradient which occurs during the execution of applications adversely affect the life-span of the system. Particularly, in 3D multi-core systems where power cannot be well balanced due to heterogenous heat dissipation, temperature gradient can be of great concern. Therefore, it is imperative that we develop solutions that can deal with temperature minimization as well as gradient reduction. Since thermal gradient is closely related to the run-time execution, after deploying our off-line thermal-aware task mapping with DVFS, we next propose our on-line optimization methodology to deal with the run-time temperature differences across the

ALGORITHM 2: Online Thermal Optimization	
Input: Application to be executed after off-line mapping and DVFS	
assignment	
Output: Thermally optimized result	
for each periodic point L_t do	
Collect temperature for all stacks and store in the queue Q ;	
repeat	
Exchange workload from hottest to coolest stack;	
if Period constraint not violated then	
Schedule workload and remove stacks from Q ;	
end	
until stacks available OR Q is not empty;	
end	

chip.

An overview of our proposed online thermal optimization technique is presented in Algorithm 2. In this approach, we employ workload migration between hot stacks and cool stacks at the beginning of a new periodic interval. At the end of each period, we collect temperature information of each stack from the thermal sensors and store it in a queue data structure in descending order. Following it, we start exchanging workload from the hottest stack to the coolest stack and vice versa without violating periodic constraint. In order to guarantee the periodic constraint we check the length of critical path identified in our off-line analysis (Equation 3) after exchanging the workload. If the length of the critical path is not greater than the length after DVFS assignment, we continue with the exchange. However, if the length is increased, we check the next cooler stack in the queue for the possibility of exchange. This approach is a very simple and light weight approach with worst case complexity of $\mathcal{O}(n^2)$, where n is the number of stacks. Due to its lighter bound, this method can sometimes results in pessimistic solutions due to task migration overheads. More sophisticated approaches can be applied to guarantee the constraint violations during workload migration, which we would consider in our future work.

After presenting our novel thermal-aware approach, we next discuss our experiments and results that validate the efficiency of our proposed algorithm.

V. EXPERIMENTS

This section presents the analysis on the performance evaluation of our proposed two-stage mapping algorithm. We begin with experimental setup for thermal simulations and profiling. Based on it, we compare our approach with some of the existing methodologies for peak temperature reduction. It is worthwhile to mention that in our experiments we assumed homogeneous 3D-MPSoC. However, our approach is independent of the type of processors. It can be easily extended to heterogeneous platforms.

A. Experimental Set Up

We model a homogeneous 3D-MPSoC as a 3X3 mesh stacked on two active layers. The cores are derived from 65nm based UltraSPARC T2 [4] processor. We assume each core can run in different modes with the supply voltage range of 0.6: 0.05: 1.0V. We calculate the corresponding frequency for each voltage level from the work presented in [16].

The peak dynamic power of the core is assumed to be 5Wwhen running at the highest mode according to the datasheet of UltraSPARC processor. Based on it, the parameter γ in



Fig. 4. Power Load Distribution on 3X3X2 Mesh 3D MPSoC

Equation 1 can be calculated. We assume in our simulation that γ is same for all the voltage modes. Then Equation 1 can be used to calculate the dynamic power for different running modes. Next, using 65nm based leakage current equation presented in [16], we computed the curve fitting constants for the linear temperature-dependent leakage power model in Equation 2 for the temperature range of $45-110^{\circ}C$. Note that we use the linear model instead of the leakage model in [16] is because that the latter is only slightly more accurate but with much higher complexity.

The thermal parameters including the area of core and memory are provided in Table I. The TSVs are implemented similar to the work presented in [4] and [5].

B. Thermal Profiling Result

As presented in Section IV-A, we performed thermal profiling on our 3D MPSoC platform. We started by collecting steady state temperature of each core when they are running on the highest mode of execution consuming dynamic power of 5W. Following it, for different peak temperature constraints, we collected the power distribution pattern over the two active layers of the 3D MPSoC. As demonstrated in Figure 4, we can see that as the temperature constraint is tighter, more and more power load is concentrated on the layer 1 (closest to heat sink). That means we need to run the cores in layer 1 at max mode for longer time, and hence completing more workload.

C. Performance Evaluation

The application model used in this work is based on the task graphs, which is a relatively new area and less explored in 3D multi-core domain for temperature management, especially for temperature reduction under throughput/periodic constraint. Therefore, in order to evaluate the efficiency of our proposed

TABLE I Thermal Simulation Parameters		
Parameter	Value	
Core area	$14mm^2$	
L2 cache area	$14.5mm^2$	
Die thickness	0.15mm	
Inter-layer material thickness	0.02mm	
Die capacitance	$1.76 \times 10^{-6} J/(m^3 \cdot K)$	
Die resistance	0.01mK/W	
Inter-layer material resistance	0.25mK/W	

off-line/on-line algorithm at different aspects, we compared our approach against two different existing approaches which are extended to match our requirement for fair evaluation. The first approach we considered is a load balancing methodology that distributes workload across the stacks as a whole with hot tasks running on lower core of the stack and cooler task on upper core (away from the heat sink). This work is an extension of the methodology presented in [25]. The second approach we considered is an extension of the methodology proposed in [11]. As a modification, to reduce complexity of implementation, we used list scheduling instead of simulated annealing to get the final solution of mapping the tasks for minimum communication cost. We selected these two approaches for the evaluation as one of the approach uses the traditional way of balancing power adapted to 3D platforms and the second approach emphasizes on communication cost under thermal constraint on 3D platforms. When comparing to these two approaches we cover both the important aspect of DAG based applications scheduled on 3D multi-core, i.e., impacts of communication and computation costs on temperature.

In the first set of experiments, we consider 50 randomly generated task graphs (*i.e.*, DAGs) in each test case. The task graphs were generated using TGFF [6]. For different periodic ratio, we evaluated the efficiency of our approach (denoted TAM-DVFS) against the load balancing approach (Pow-bal) [25] and communication aware approach (Com-A) [11] and the results are shown in Figure 5. From the figure, we can see that, our proposed approach outperforms both the approaches and achieves average peak temperature reduction up to $14^{\circ}C$. In this experiment, we employ DVFS to our approach alone to demonstrate the effectiveness of our DVFS assignment methodology in reducing the peak temperature. This might raise the question that the temperature reduction is due to DVFS only.

To investigate this query, we performed the similar experiment again, but this time we enabled the other two approaches with DVFS. Figure 6 demonstrates the results of this experiment, which clearly shows that even after applying DVFS to other approaches, our approach is still more efficient and can reduce more average peak temperature up to $5.5^{\circ}C$ than the other two. The result in Figure 6 validates that our thermalaware task mapping algorithm that uses wise decisions based on thermal profiling is effective and provide better opportunity in peak temperature reduction. Note that these two figures are generated based on different benchmark sets, but reflect the same trend.

To further demonstrate the scalability of our approach, we run a new set of experiment in which we fixed the periodic ratio and increase the number of nodes in each application. By doing this experiment, we evaluate the efficiency of our approach when there is limited opportunity to use the slack time available for temperature reduction. The result of this experiment are shown in Figure 7. From Figure 7, we can see that even with increasing nodes, our approach can reduce peak temperature up to $4.35^{\circ}C$ compared to the other two approaches.

Furthermore, we also demonstrate that our proposed approach is highly flexible and can achieve similar results with extended to more layers. We run a similar experiment with



Fig. 6. Impact of Increasing Periodic Ratio(DVFS)

increasing number of nodes on a new 3D platform with 3X3 mesh on 3 layers. The trend of our result was similar as in Figure 7.

VI. CONCLUSION

In this paper, we propose a novel light weight solution to thermal challenges faced by modern day 3D multi-core systems. Our approach is a two-stage approach that performs thermal-aware task mapping and DVFS assignment at design time, and light weight simple thermal optimization step at run-time. Our proposed approach outperforms other existing methods and achieves average peak temperature reduction up to $14^{\circ}C$.

ACKNOWLEDGEMENT

This work was supported by A*Star - SERC Public Sector Funding (PSF) of Singapore under Grant No. 1121202015.

REFERENCES

- [1] Hotspot 5.02 temperature modeling tool. *University of Virgina*, page http://lava.cs.virginia.edu/HotSpot, 2011.
- [2] C. Addo-Quaye. Thermal-aware mapping and placement for 3-d noc designs. In *IEEE SoC Conference (SOCC)*, pages 25–28, 2005.



Fig. 7. Impact of Increasing Task Nodes

- [3] A. K. Coskun, J. L. Ayala, D. Atienza, T. S. Rosing, and Y. Leblebici. Dynamic thermal management in 3d multicore architectures. In *Proc. IEEE Design, Automation* & *Test in Europe (DATE) Conference*, pages 1410–1415, 2009.
- [4] A. K. Coskun, A. B. Kahng, and T. S. Rosing. Temperature and cost-aware design of 3d multiprocessor architectures. In *Proc. Euromicro Conference on Digital System Design (DSD)*, pages 183–190, 2009.
- [5] M. Cox, A. K. Singh, A. Kumar, and H. Corporaal. Thermal-aware mapping of streaming applications on 3D Multi-Processor Systems. In *Proc. IEEE Symposium on Embedded Systems for Real-time Multimedia (ESTIMedia)*, pages 11–20, 2013.
- [6] R. P. Dick, D. L. Rhodes, and W. Wolf. Tgff:task graph for free. In Proc. International Workshop on Hardware/Software Codesign (CODES/CASHE), pages 97–101, 1998.
- [7] J. U. Knickerbocker et al. 3-d silicon integration and silicon packaging technology using silicon through-vias. *IEEE Journal of Solid-State Circuits (JSSC)*, 41(8):1718– 1725, 2006.
- [8] M. S. Bakir et al. 3D heterogeneous integrated systems: Liquid cooling, power delivery, and implementation. In Proc. IEEE Custom Integrated Circuits Conference (CICC), pages 663–670, 2008.
- [9] P. Zhou et al. 3d-staf: scalable temperature and leakage aware floorplanning for three-dimensional integrated circuits. In Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pages 590–597, 2007.
- [10] T. Chantem et al. Temperature-aware scheduling and assignment for hard real-time applications on mpsocs. *IEEE Transactions on VLSI Systems (TVLSI)*, 19(10): 1884–1897, 2011.
- [11] Y. Cheng et al. Thermal-constrained task allocation for interconnect energy reduction in 3-d homogeneous mpsocs. *IEEE Transactions on VLSI Systems (TVLSI)*, 21(2):239–249, 2013.
- [12] S. Feero and P. P. Pande. Networks-on-chip in a three-dimensional environment: A performance evaluation. *IEEE Transactions on Computers (TC)*, 58(1):32– 45, 2009.
- [13] Y. K. Kwok and I. Ahmad. Static scheduling algorithms for allocating directed task graphs to multiprocessors.

ACM Journal of Computing Surveys (CSUR), 31(4):406–471, 1999.

- [14] J. H. Lau and T. G. Yue. Thermal management of 3D IC integration with TSV (through silicon via). In *Proc. IEEE Electronics Packaging Technology Conference (EPTC)*, pages 635–640, 2009.
- [15] J. Li, M. Qiu, J-W Niu, L. T. Yang, Y. Zhu, and M. Zhong. Thermal-aware task scheduling in 3D chip multiprocessor with real-time constrained workloads. *ACM Transaction on Embedded Computing (TECS)*, 12 (2):24, 2013.
- [16] W. Liao, L. He, and K.M. Lepak. Temperature and supply voltage aware performance and power modeling at microarchitecture level. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 24(7):1042 – 1053, 2005.
- [17] S. Liu, J. Zhang, Q. Wu, and Q. Qiu. Thermal-aware job allocation and scheduling for three dimensional chip multiprocessor. In *Proc. International Symposium on Quality Electronic Design (ISQED)*, pages 390–398, 2010.
- [18] R. S. Patti. Three-dimensional integrated circuits and the future of system-on-chip designs. *Proc. IEEE*, 94(6): 1214–1224, 2006.
- [19] K. Puttaswamy and G. H. Loh. Thermal herding: Microarchitecture techniques for controlling hotspots in high-performance 3d-integrated processors. In Proc. International Symposium on High-Performance Computer Architecture (HPCA), pages 193–204, 2007.
- [20] G. Quan and V. Chaturvedi. Feasibility analysis for temperature- constraint hard real-time periodic tasks. *IEEE Transaction on Industrial Informatics*, 6(3):329– 339, 2010.
- [21] J. M. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits: A Design Perspective*. Prentice Hall, 2003.
- [22] C. Sun, L. Shang, and R. P. Dick. Three-dimensional multiprocessor system-on-chip thermal optimization. In *Proc. International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pages 117–122, 2007.
- [23] E. Wong and S. K. Lim. 3D floorplanning with thermal vias. In *Proc. IEEE Design Automation and Test in Europe (DATE)*, volume 1, pages 1–6, 2006.
- [24] H. Zhao and R. Sakellariou. Scheduling multiple dags onto heterogeneous systems. In *Proc. IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 159–173, 2006.
- [25] X. Zhou, Y. Xu, Y. Du, Y. Zhang, and J. Yang. Thermal management for 3d processors via task scheduling. In *Proc. IEEE International Conference on Parallel Processing (ICPP)*, pages 115–122, 2008.
- [26] X. Zhou, J. Yang, Yi. Xu, Y. Zhang, and J. Zhao. Thermal-aware task scheduling for 3d multicore processors. *IEEE Transactions on Parallel and Distributed Systems(TPDS)*, 21(1):60–71, 2010.
- [27] C. Zhu, Z. Gu, L. Shang, R. P. Dick, and J. Russ. Three-dimensional chip-multiprocessor run-time thermal management. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 27 (8):1479–1492, 2008.