Fast Algorithms for Simultaneous Optimization of Performance, Energy and Temperature in DAG Scheduling on Multi-Core Processors

Hafiz Fahad Sheikh[1] **and Ishfaq Ahmad**[1]
[1]Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX, USA

Abstract - *We have proposed two algorithms for simultaneously optimizing performance, energy, and temperature while scheduling a set of tasks on a multi-core system (PET-Scheduling). The proposed algorithms differ in the way they use task allocation and voltage selection decisions to obtain multiple schedules (trade-off solutions) with a wide range of values along each objective. PET-PPF combines a power and performance-aware allocation scheme with probabilistic voltage selection to obtain these trade-off solutions. PET-DCP, on the other hand, first adjusts the task execution times using probabilistic voltage selection before leveraging a performance-optimal scheduler to generate the final schedule. Our results on several application task graphs demonstrate that both of the proposed algorithms can obtain the trade-off curves comprising of multiple solutions to the PET-Scheduling problem. PET-DCP, however, is able to achieve identical energy and thermal improvements as that of PET-PPF but at the same time degrades the performance by as much as 1.5 times less than that of the PET-PPF.*

Keywords: dynamic thermal management, frequency allocation, multi-core systems, task scheduling, DVFS.

1 Introduction

The rapidly scaling multi-core architecture has already spanned to 100 cores on a single chip [20]. However, this quick performance gain has resulted into the complex problems of energy and thermal management. Amplified chip temperatures not only require extra efforts for cooling in the form of expensive thermal packaging or larger fan size but can also lead to problems which can affect lifespan, reliability and performance of these modern systems. For example, a higher temperature can degrade the lifespan of a system to half of its value with a nearly 10^{o}C increase in the operating temperature [2]. It has also been found that in addition to the higher temperatures, large magnitude of thermal gradients can also adversely affect the performance of the interconnects and thus can limit the performance of a system [3]. Therefore, scheduling schemes that can help to control or maintain the temperature below a given threshold are an essential requirement for the extensive use of these systems. A lot of research in the last few years has focused on temperature management, temperature-aware scheduling and performance issues related to these schemes [5], [9], [10]. Most of the research contributions target to satisfy an energy budget or a thermal constraint while minimizing the consequential performance degradation [1], [11]. While these schemes can serve to meet the imposed system-based constraints, they are unable to explore the best possible trade-off between performance and energy or performance and temperature. In addition, some important issues cannot be addressed by constraint-satisfying approaches. For example, for a certain margin of trade-off in performance, what are various improvements possible in energy and thermal profile? For different possible energy budgets and thermal constraints, what is the maximum value of performance that can be achieved? Given a set of schedules with varying values of performance, energy, and temperature, how to select the best trade-off solution? These questions demand a holistic approach for integrating performance (*P*), energy (*E*), and temperature (*T*) (*PET quantities*) into the scheduling process. For this, we address the problem of simultaneously optimizing performance, energy, and temperature while scheduling tasks on a multi-core system (*PET-Scheduling*). Such joint optimization of performance, energy, and temperature is not only complex and challenging but is also a rather unexplored problem.

PET-Scheduling problem is an aggregate of task allocation, task scheduling, and voltage selection problems with the goal of minimizing the performance, energy and temperature. We have developed novel algorithms namely PET-PPF and PET-DCP that can judiciously trade-off performance with energy and temperature. PET-PPF generates a set of probability distributions for selecting a voltage level for each task. Each distribution corresponds to a different value of expected voltage level and thus enables PET-PPF to obtain several trade-off solutions. PET-DCP, uses the same voltage selection scheme as PET-PPF, however, a significant difference is that PET-DCP performs the voltage selections before the task allocation phase. In contrast to PET-PPF, PET-DCP first updates the execution times of the tasks based on the corresponding voltage level selected for each task and then uses a performance-optimal scheduler (DCP [19]) to generate the complete schedule. The key strength of our proposed methods is that they do not aim to provide a single solution to the problem. Rather several trade-off solutions are determined for the *PET-Scheduling* problem. This approach to the *PET-Scheduling* problem is correct as in the presence of multiple conflicting objectives one solution can dominate the other along

different objectives. And for such cases, where all quantities are equally important, one quantity cannot be directly preferred over the other.

The rest of the paper is organized as follows: Section II covers the related work on energy and thermal-aware scheduling. Section III presents the details of the problem under consideration. Section IV explains the proposed algorithms for the solution of the problem and Section V highlights the evaluation setup. Section VI explains the results of the simulation while Section VII concludes the paper.

2 Related Work

Most of the research efforts in the energy and thermal-aware scheduling, target to satisfy a given thermal or energy constraint at the cost of some loss in performance. Primarily, dynamic voltage and frequency scaling (DVFS) is used to adjust the voltage levels of the cores to reduce the power consumption and thus the energy and temperature. The methods in [6], [9], [10], [15] aim to meet the thermal constraints for different kinds of workloads and systems under consideration. A solution for maximizing performance under the imposed power and thermal constraints, by solving the frequency assignment problem for multi-core systems is presented in [9]. Another frequency planning method leveraging combinatorial optimization framework to maximize performance of multi-core systems with thermal limits is developed in [10]. An Event based scheduling method that can improve the peak temperature along with the total number of DTMs without excessive computation overhead is presented in [15]. The scheme proposed in [6] uses a non-DVFS approach to calculate the optimal core states for the given thermal constraints. For tasks represented as DAGs (Directed Acyclic Graphs), several iterative techniques in terms of performance degradation and computational complexity are compared in [11]. These techniques aim to find the most suitable task for the voltage adjustment to satisfy the given thermal constraint. There are also several methods that handle DTM with an alternate perspective, instead of considering temperature as a form of constraint, they try to improve thermal profile of the system within the allowed performance margins [5], [8], [14]. Similarly, there are numerous research efforts which aim to meet the given energy budget while sacrificing as little performance as possible. A power shifting method in [25] that controls the power of the different components of a server system is shown to improve the power budget based on workload conditions. A method to minimize schedule length under an energy constraint by determining the optimal power supplies on each processor is outlined in [18].

However, there are far lesser number of contributions in the pursuit of joint optimization of performance and energy or performance and temperature. The scheme presented in [4] minimizes the energy consumption and performance penalty leveraging Complier-Driven techniques. A hybrid hardware-software approach can improve performance loss consequential to the application of DVFS by leveraging reconfigurable fetch, issue, and retirement units etc., without exceeding the thermal limit [12]. The impact on the thermal profiles of cores due to the power activation at different locations of the chip represented as look-up tables have been used in [16] to allocate tasks to different cores. The proposed approach targets to improve the peak temperature as well as guarantee the thermal limit while at the same time decreases the rejection ratios under thermally constrained CMPs. A few research efforts report improvements in performance and energy under an efficient DTM policy [13]; however, these improvements are usually the by-products of the thermal management.

In contrast to the above mentioned research contributions we have targeted the simultaneous optimization of performance, energy, and temperature for allocating tasks on a multi-core system. We have compared our solutions to the schedules which only target to achieve maximum performance and do not take energy and temperature into consideration. This comparison can help to quantify the actual performance loss exhibited by various solutions in the pursuit of energy and temperature minimization.

3 PET-Scheduling Problem

Given a task graph with N tasks, the total number of cores (M) and the set of available voltage levels ($\boldsymbol{L}$), we aim to minimize makespan, energy consumption and temperature simultaneously while solving the task allocation, task scheduling, and voltage selection problem for the given task set. Thus the required objectives are:

$$Minimize \quad \max_{1 \le i \le N} ft_i \tag{1}$$

$$Minimize \quad \sum_{i=1}^{N} P_i.et_i \tag{2}$$

$$Minimize \quad \max_{1 \le i \le N} \max_{1 \le j \le M} T_i^j \tag{3}$$

ft_i represents the finish time of the ith task. P_i is the power dissipated during the execution of the ith task and et_i is corresponding execution time. T_i^j is the temperature of the jth core during the execution of the ith task (details of thermal model will be presented in Section V). Our goal is not to only produce one solution for solving the above mentioned min-min-min problem but to explore the whole Pareto front that exists between performance, energy, and temperature (*PET quantities*). These trade-off solutions can be used to guide the overall scheduling process to meet the required objectives. For workload, we considered tasks with precedence relationships represented as directed acyclic graphs (DAGs). Several scientific and multimedia applications can be conveniently represented as DAGs. A DAG consists of weighted nodes and edges, where the

weight of the node represents the cost associated with the computation of the task and the weight on an edge represents the communication cost between the two tasks or nodes. Critical path in a DAG is defined as the path of longest length in the graph and hence governs the latest finishing time of a scheduled DAG [7]. The nodes constituting critical path are known as critical path nodes (CPNs) and the cores on which these tasks are scheduled are called the critical cores. Nodes having successors on critical path are known as In-bound nodes (IBNs). All other nodes are called out-bound nodes (OBNs) [7].

4 Proposed Solution

We will explain the algorithms proposed for solving the *PET-Scheduling* problem by highlighting their task allocation and voltage selection phases followed by their computational complexities. While doing so, it is assumed that there are *M* cores in the system which can switch across *K* voltage levels and the DAG to be scheduled has *N* tasks/nodes.

4.1 PET-PPF

For DAG scheduling, the decision space for the *Pet-Scheduling* problem spans not only the task allocation decisions but also include task ordering and voltage selection decisions. PET-PPF solves the problem in a hierarchical manner. For task allocation, PET-PPF aims to minimize the product of total power consumption of the cores and their available time for allocating the upcoming task. The intuition is to include performance and power directly into the allocation decisions as both energy consumption and temperature are related to the power dissipation. In other words, while allocating ith task we select the core with minimum PP^j_i which is defined as:

$$PP_i^j = s_{i-1}^j \, P_{i-1}^j, \quad \forall 1 \le j \le M \tag{4}$$

where, s_{i-1}^j represents the finish time of the jth core after allocating i-1 tasks to all the cores. P_{i-1}^j is the total power consumption of the jth core just before allocating ith task. Therefore, ith task is allocated to the core such that:

$$y_{i,j} = \begin{cases} 1 & for\ j = \arg\min_{1 \le \delta \le M} (PP_i^\delta) \\ 0 & otherwise \end{cases} \tag{5}$$

In (5), $y_{i,j}$ is set to 1 if ith task is allocated to the jth core. The makespan of a scheduled DAG varies significantly with how tasks are prioritized during the allocation phase. We assigned priorities to the tasks according to their classification in the DAG. CPNs are given the highest priority followed by IBNs while OBNs are kept at the lowest priority. The list of tasks generated by this classification is usually termed as CPN Dominant Sequence (CPN-DS) [7]. However, while constructing CPN-DS the precedence constraints are evaluated to ensure that the parent nodes are added to the list prior to the task itself. In voltage selection

PET-PPF

```
1: K=Total number of voltage levels
2:Initialize P_dist ← uniform distribution for voltage selection
3:// First Shifting the peak towards highest voltage level
4:pivot_ point ← select a partition point on the set of voltage levels
5:for given number of adjustment steps (τ)
6:   for all levels from start to pivot_point
7:      reduction ← P_dist(level)  / reductionPerStep (η)
8:      P_dist(level)  ← P_dist(level)  -reduction
9:      creduction+= reduction
10:  end for
11:  for all levels from pivot_point to K
12:     P_dist(level) = P_dist(level)+ proportional factor*creduction
13:     // Higher voltage levels get large components of creduction
14:  end for
15:  Vlevels =generateVlevels( P_dist)
16:  for all tasks ϵ DAG (V, E)
17:     selected_core ← find the core with minimum TP_product
18:     Allocate task to selected_core at the earliest possible time ST
19:     udateSystemState();
20:  endfor
21:  updateSolutionSet(currentSchedule);
22:end for
23:Initialize P_dist ← 1/K
24:// Shifting the peak towards lowest voltage level
25:for given number of adjustment steps (τ)
26:  for all levels from pivot_point to K
27:     reduction ← P_dist(level)  / reductionPerStep (η)
28:     P_dist(level)  ← P_dist(level)  -reduction
29:     creduction+= reduction
30:  end for
31:  for all levels from start to pivot_point
32:     P_dist(level) = P_dist(level)+ proportional factor*creduction
33:     // Lower Voltage levels get large components of creduction
34:  end for
35:  Vlevels =generateVlevels( P_dist)
36:  for all tasks ϵ DAG (V, E)
37:     selected_core ← find the core with minimum TP_product
38:     Allocate task to selected_core at the earliest possible time ST
39:     updateSystemState();
40:  endfor
41:  updateSolutionSet(currentSchedule);
42:endfor
```

Figure 1: PET-PPF

phase, a set of probability distributions is generated. Each probability distribution is used to select the voltage level for every task in the DAG, thus generating a potentially different schedule in the objective domain. In other words, to obtain τ trade-off solutions, we generate τ probability distributions. For generating this set of distributions, we start with a uniform distribution (allowing each voltage level to have the equal chance of getting selected for every task). We then transform the distribution in each step to first shift the peak of distribution towards the maximum voltage level and then repeat the procedure starting with uniform distribution to shift the peak of distribution towards the lowest available voltage level. In other words, we start with a uniform probability distribution in first step, as:

$$\Pr(x = L_i) = \frac{1}{K}, \ \forall L_i \in \mathbf{L} \tag{6}$$

In the above equation **L** represents the set of available voltage levels. Now, to adjust the peak of this distribution we define a *partition index* on the set **L**. Such that in each

PET-DCP

```
1:L=loadVoltageLevels
2:distributionData=generateProbDist()
3:for n=1 to size(distributionData)
4:   currentprob=distributionData(n);
5:   vLevels=generateVlevels(currentprob);
6:   currenttaskgraph =updateTaskGraph(filename,currentprob);
7:   newshedule=dcpSchedule(currenttaskgraph);
8:   makespan=getLatestCompletiontime(newschedule);
9:   energyconsumption=
10:  getEnergyConsumption(newschedule,vLevels)
11:  maxTemp=max(getThermalProfiles(newschedule,vLevels))
12:  updateSolutionSet(currentSchedule);
13:endfor
```

Figure 2**:** PET-DCP

transformation step the probabilities corresponding to the voltage levels with indexes up to the *partition index* are reduced by a certain factor and the collective reduction is then distributed to the probabilities of voltage levels present in the second partition. If α represents the *partition index* over the set of available voltage levels and η defines the fractional reduction in the probability values corresponding to the selected voltage levels. Then the new distribution can be given by:

$$\Pr(L_i)=\begin{cases}\Pr(x=L_i)/\eta & \forall\, i\le\alpha \\ \Pr(x=L_i)+\dfrac{L_i}{\sum_{r=\alpha}^{K}L_r}\sum_{m=1}^{\alpha}(\eta-1)\Pr(x=L_m) & \forall\ \alpha\le i\le K\end{cases} \qquad (7)$$

In the above equation the parameters α and η can be adjusted to control the computational complexity of the overall approach. We found empirically that with α set to the middle of set **L** and η=2, several unique distributions can be obtained when size of **L** is not very large ($|\mathbf{L}| \le 10$). It must be noted that (7) represents the adjustments done for gradually shifting the probability distribution to favor the maximum voltage level and may be repeated τ times, where τ can be selected based on the value of η. In addition, the increase in the probability of selection for each level is proportional to the value of its voltage. The required modification is straight forward for the case where we need to favor the lowest voltage level. Figure 1 presents the overall procedure used by the PET-PPF.

4.1.1 Computational Complexity of PET-PPF

The total number of adjustments steps can be controlled by τ for each direction, therefore, the probability distribution adjustment phase is $O(\tau K)$. In the task allocation phase, a single term as in (4) is evaluated for every core per each task. Hence, the complexity of task allocation phase is $O(M)$. So, for a DAG with N tasks, the overall complexity of PET-PPF is $O(\tau N (M+K))$.

4.2 PET-DCP

PET-DCP takes an opposite approach to that of the conventional DTM and energy improvement schemes. Most of such schemes adjust a performance-optimal schedule to satisfy the given thermal and energy requirements. However, PET-DCP, starts with the task adjustment phase using the given trade-off margin, and then leverages a performance-optimal scheduler to generate the final schedule. In other words, initially a set of probability distributions (similar to PET-PPF) for voltage selection phase is generated. Each distribution is then used to select the voltage level for every task in the DAG. Based on the selected voltage levels, the execution time of each task is updated. This updated DAG is then used as input to the DCP (Dynamic Critical Path) scheduler [21] which generates the final schedule. DCP is a performance-optimal scheduler which keeps track of the critical path after every task allocation and assigns priorities to the remaining tasks accordingly. DCP has been shown to generate schedules with near-optimal makespan [7]. As the expected voltage level varies across the probability distributions generated in the voltage selection phase. Therefore, each distribution allows a different level of performance trade-off which translates into possibly different energy and thermal improvements. It should be noted that any change in the execution time of tasks can result into a possible modification of the critical path and thus the initial schedule as generated by DCP no longer remains optimal. Therefore, starting from a performance-optimal schedule and then iteratively updating it for the desired energy and thermal requirements may potentially lose more performance in the pursuit of energy and temperature improvements. However, the use of performance-aware scheduler by PET-DCP as a second step ensures maximum performance for the modified task graph. Thus, PET-DCP can potentially achieve the performance-energy and performance-temperature trade-offs without excessive performance degradation. Figure 2 briefly outlines the PET-DCP approach.

4.2.1 Computational complexity of PET-DCP

The computational complexity of DCP to generate a schedule for a DAG of N tasks is $O(N^3)$. The total number of adjustments steps are 2τ, therefore the complexity of probability distribution adjustment phase is $O(\tau K)$. Ignoring the time-complexity of drawing N levels from the given distribution ($O(KN)$) and the time to update the DAG ($O(N)$), the overall complexity of PET-DCP will be $O(\tau K N^3)$.

5 Experimental Details

We assumed a 16-core system with cores arranged in a grid layout of 4x4. However, the proposed approach can be used for any number of cores and voltage levels. Each core was assumed to be able to switch across 5 different voltage levels in active mode, thus changing the power consumption and frequency of the system. The values of frequencies at different voltage levels along with their power consumption are outlined in Table 1. It should be noted that the frequency-power scaling relationship used in our evaluation

TABLE 1
DVFS PARAMETERS

f(MHz)	1600	2000	2200	2400	2600
P(W)	23.61	48.90	72.48	93.12	105.00

TABLE 2
SYSTEM PARAMETERS

Parameter	No. of Cores	Layout	Freq. Switching	Partition index (α)	Adj. factor/step (η)	Total adj. steps (τ)	Total no. of solutions
Value	16	Grid 4x4	Independent	3	2	10	27

is not very aggressive in terms of reducing the power with the change in frequency/voltage level. Similar scaling relationships have been observed by other research efforts based on actual multi-core systems [17]. Various other parameters related to the system under consideration and the proposed algorithms are listed in Table 2.

5.1 Thermal model

To estimate the temperature of the cores with various power dissipation levels, we can use a steady state thermal model as:

$$T_j = R_{th}\, P_j + T_A \tag{8}$$

In the above equation, T_j represents the temperature of the jth core due to a power dissipation of P_j watts. R_{th} is the thermal resistance and T_A represents the ambient temperature. Though the model in (8) has been frequently used in various DTM related research efforts, however, it does not take into account the power consumption of the neighboring cores while calculating the temperature of each core. In order to cater for the power dissipation of the neighboring cores we can modify (8), similar to [24] as:

$$T_j = R_{th} P_j + \sum_{\forall m \in neighbor_j} \gamma R_{th}\, P_m + T_A \tag{9}$$

In (9), $neighbor_j$ represents the set of cores which are adjacent to the jth core. The correlation between the power consumption of the neighboring cores and the temperature of a particular core can be controlled by γ. Though the model in (9) is still simplistic, however, it should be noted that any thermal and system model can be used in conjunction with the proposed algorithms. Since, the presented algorithms provide a mechanism to explore trade-off surfaces that exist between performance, energy, and temperature, therefore, the values of these quantities can be obtained from any complex and detailed models without having an impact on the results reported in Section VI.

5.2 Task model

We used task graphs of various applications including Fast Fourier Transform [22], Laplace Equation [23], Gauss Elimination [22], Fpppp [21] and a Robot Control application [21]. Details of these task graphs can be found in the provided references.

6 Results

Figures 3-5 compare the trade-off regions obtained by the proposed algorithms for various application task graphs. While figures 6-8 present the corresponding performance-

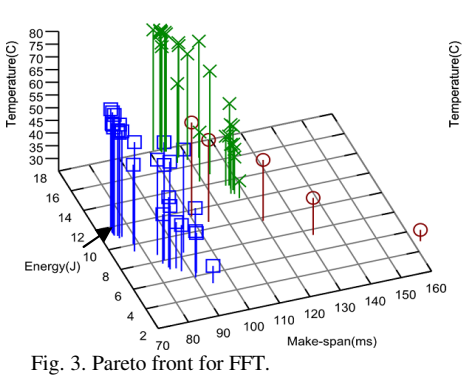

Fig. 3. Pareto front for FFT.

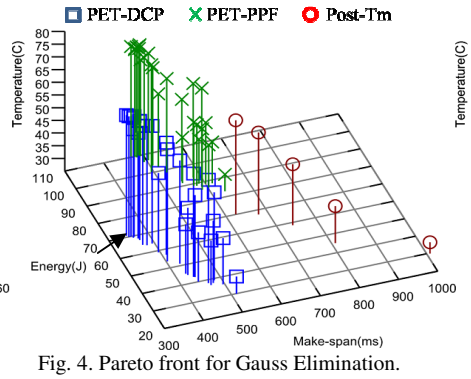

Fig. 4. Pareto front for Gauss Elimination.

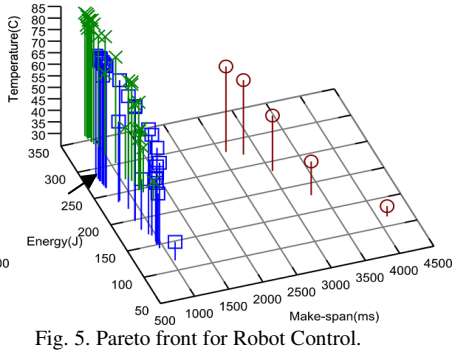

Fig. 5. Pareto front for Robot Control.

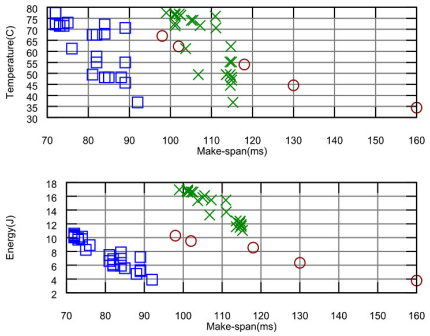

Fig. 6. Trade-off curves for FFT.

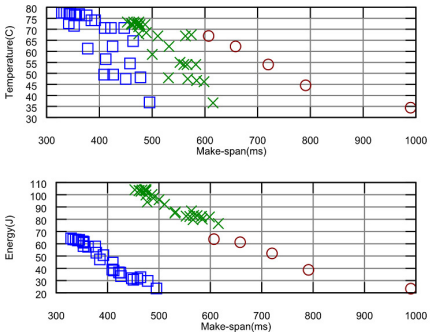

Fig. 7. Trade-off curves for Gauss Elimination.

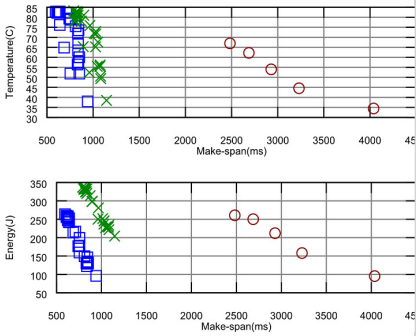

Fig. 8. Trade-off curves for Robot Control.

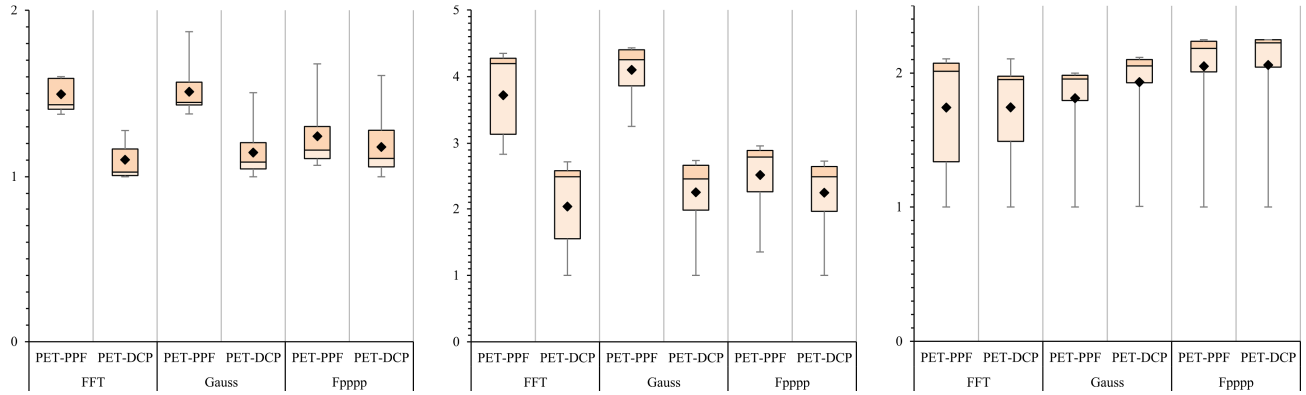

Fig. 9. Comparison of PET-PPF and PET-DCP for various task graphs along (a) Performance, (b) Energy, and (c) Peak temperature.

energy and performance-temperature trade-offs possible by leveraging the Pareto-fronts obtained from the proposed algorithms. Due to the space considerations, we have only presented the selected figures and tables. For comparison, we used an efficient temperature-aware allocation scheme called Post-Tm [16]. We used each of the available voltage level from Table 1 along with the Post-Tm allocation scheme to generate a base-line trade-off surface. A direct comparison of these Pareto surfaces (Figures 3-8) proves that both of the proposed algorithms can obtain trade-off solutions with wide range of values and with better spread along each objective as compared to the modified Post-Tm approach. However, a contrasting difference is that PET-DCP is able to generate the Pareto fronts or trade-off surfaces which are much closer to the performance optimal point (shown by arrows in Figures 3-5). Since, the spread of solutions generated by the PET-DCP is not inferior to both PET-PPF and Post-Tm along energy and temperature axis, therefore, the closeness to origin along performance axis translates into energy-performance and temperature-performance trade-offs with lesser performance degradation. Figure 9 compares the algorithms in terms of the distribution of the solution points (normalized to the minimum value obtained) along each objective. We can observe that PET-DCP was able to achieve a range of values comparable to PET-PPF but on the other hand, attains a significantly lower mean value for most of the cases.

To further analyze the quality of trade-offs, Table 3 compares the amount of performance degradation for the corresponding improvements in energy and temperature for each algorithm. While calculating the performance degradation as well as energy and temperature reductions for each trade-off solution, the values of *PET quantities* corresponding to the performance-optimal schedule as generated by DCP were used as reference. $\Delta P_m/\Delta E_m$ represents the ratio of percentage performance degradation to the percentage decrease in energy and $\Delta P_m/\Delta T_m$ is the ratio of percentage performance degradation to the percentage reduction in peak temperature (averaged over all solutions generated by each algorithm for a given task graph). Negative values for $\Delta P_m/\Delta E_m$ and $\Delta P_m/\Delta T_m$ in Table 3 represent an average decrease in the energy and temperature consequential to the corresponding performance degradation. From the values in Table 3, it can be observed that both the PET-PPF and Post-Tm degrades the performance by a larger percentage than the corresponding percentage decrease in energy and temperature. For example, for Gauss Elimination task graph the values of $\Delta P_m/\Delta T_m$ for PET-PPF and Post-Tm are -3.04 and -3.99 which means that for every 1% reduction in peak temperature, performance has to be degraded by 3.04% and 3.99% respectively. However, PET-DCP needs to degrade the performance only by 1.43% for every 1% improvement in peak temperature for the same application. Similar comparison exists for other task graphs. Figure 10 highlights this trend pictorially for all the task graphs used in our experiments. We also observe that PET-PPF, yields positive values for $\Delta P_m/\Delta E_m$ for some of the task graphs. This points out that, PET-PPF is unable to obtain large number of trade-off solutions that can improve energy consumption as

TABLE 3
TRADE-OFF RATIOS

Application	PET-PPF		Post-Tm		PET-DCP	
	$\Delta P_m/\Delta E_m$	$\Delta P_m/\Delta T_m$	$\Delta P_m/\Delta E_m$	$\Delta P_m/\Delta T_m$	$\Delta P_m/\Delta E_m$	$\Delta P_m/\Delta T_m$
Fft	1.11	-2.89	-2.96	-2.13	-0.49	-0.59
Gauss	1.31	-3.04	-5.00	-3.99	-0.77	-1.43
Laplace	1.75	-2.03	-5.29	-4.16	-0.90	-1.36
Robot	7.89	-3.84	-16.08	-11.40	-0.87	-1.85
Fpppp	-2.05	-2.56	-28.27	-17.82	-0.94	-2.07
Average	2.00	-2.87	-11.52	-7.90	-0.79	-1.46

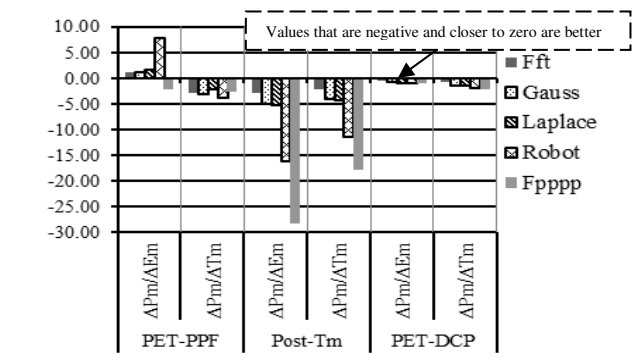

Fig. 10. Comparison of Trade-off Ratios.

compared to the base-line performance-optimal schedule.

Based on the performance of PET-DCP, we can note that in general, it may be potentially better to pre-adjust the tasks according to the system's state and the given requirements of energy and temperature before targeting the maximal performance to obtain better trade-off solutions. It may also be noted that once the trade-off solutions are available, an operating point or schedule can be selected from them according to the imposed constraints and the given preferences.

7 Conclusion

We proposed two schemes to explore the trade-off regions that may exist while trading loss in performance with the improvements in energy and peak temperature. Both algorithms were able to generate trade-off curves comprising of schedules that result into diverse range of values for makespan, energy consumption and peak temperature. Our evaluation results indicate that PET-DCP, which pre-adjusts tasks probabilistically for energy and thermal improvements before using a performance-optimal scheduler, can produce multiple schedules that are very close to the performance-optimal point. This leads the PET-DCP to achieve *trade-off ratios* better than the other algorithms (PET-PPF and Post-Tm) by a factor of 2 on average. The work presented in this paper is an inaugural effort to jointly optimize performance, energy, and temperature while scheduling tasks on a multi-core system and can be used as a framework to attain efficient trade-offs among the *PET quantities*.

8 References

[1] D. King, I. Ahmad, H.F. Sheikh, "Stretch and compress based re-scheduling techniques for minimizing the execution times of DAGs on multi-core processors under energy constraints," *Green Computing Conference, 2010 International*, pp.49-60, 15-18 Aug. 2010

[2] R. Viswananth, V. Wakharkar, A. Watwe, and V. Lebonheur, "Thermal Performance Challenges from Silicon to Systems," *Intel Technol. J., Q3,* vol. 23, p. 16, 2000.

[3] A. H. Ajami, K. Banerjee, and M. Pedram, "Modeling and Analysis of Nonuniform Substrate Temperature Effects on Global Ulsi Interconnects," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 24, no. 6, pp. 849-861, Jun. 2005.

[4] U. Kremer, J. Hicks, and J.M. Rehg, "Compiler-Directed Remote Task Execution for Power Management," Workshop on Compilers and Operating Systems for Low Power (COLP), 2000.

[5] N. Fisher, J. Chen, S. Wang, and L. Thiele, "Thermal-Aware Global Real-Time Scheduling on Multicore Systems," *15th IEEE Real-Time and Embedded Technology and Applications Symposium*, April 2009.

[6] R. Rao and S. Vrudhula, "Efficient Online Computation of Core Speeds to Maximize the Throughput of Thermally Constrained Multi-core Processors," Proceedings of the International Conference on Computer-Aided Design, November 2008.

[7] Y-K. Kwok and I. Ahmad, "Static scheduling algorithms for allocating directed task graphs to multiprocessors," *ACM Comput. Surv.* vol 31, no. 4, pp. 406-471, 1999.

[8] A. Coşkun, T. Rosing, K. Whisnant, and K. Gross, "Static and Dynamic Temperature-Aware Scheduling for Multiprocessor SoCs," *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, Vol. 16, No. 9, September 2008.

[9] T. Chantem, R. Dick, and X. Hu, "Temperature-Aware Scheduling and Assignment for Hard Real-Time Applications on MPSoCs," *Proceedings of the Conference on Design, Automation and Test in Europe*, March 2008.

[10] A. K. Coskun, T. S. Rosing, and K. C. Gross, "Proactive Temperature Management in MPSoCs," *Proceeding of the 13th international Symposium on Low Power Electronics and Design, ISLPED '08*, August 2008.

[11] H.F. Sheikh, I. Ahmad, "Fast algorithms for thermal constrained performance optimization in DAG scheduling on multi-core processors," *Green Computing Conference and Workshops (IGCC), 2011 International* , pp.1-8, July 2011.

[12] O. Khan and S. Kundu, "Hardware/Software Co-design Architecture for Thermal Management of Chip Multiprocessors," *Conference & Exhibition on Design, Automation & Test in Europe, DATE '09*, April 2009.

[13] T. Ebi, M. Faruque, and J. Henkel, "TAPE: Thermal-aware Agent-based Power Economy Multi/many-Core Architectures," *IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers, ICCAD 2009*, pp.302-309, November 2009.

[14] D. Puschini, F. Clermidy, P. Benoit, G. Sassatelli, and L. Torres, "Temperature-Aware Distributed Run-Time Optimization on MP-SoC Using Game Theory," *Symposium on VLSI, 2008. ISVLSI '08. IEEE Computer Society Annual*, pp.375-380, April 2008.

[15] J. Cui, D.L. Maskell, "High Level Event Driven Thermal Estimation for Thermal Aware Task Allocation and Scheduling," *15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2010, pp.793-798, 18-21 Jan. 2010.

[16] J. Cui and D. L. Maskell, "Dynamic Thermal-Aware Scheduling on Chip Multiprocessor for Soft Real-Time Systems," Proceedings of the 19th ACM Great Lakes Symposium on VLSI, GLSVLSI '09, May 2009.

[17] R. Cochran, C. Hankendi, A. Coskun, S. Reda, "Identifying the optimal energy-efficient operating points of parallel workloads," *Computer-Aided Design (ICCAD), 2011 IEEE/ACM International Conference on* , pp.608-615, Nov. 2011.

[18] K. Li, "Performance Analysis of Power-Aware Task Scheduling Algorithms on Multiprocessor Computers with Dynamic Voltage and Speed," IEEE Transactions on Parallel and Distributed Systems, pp. 1484-1497, November, 2008.

[19] Y-K. Kwok, I. Ahmad, "Dynamic critical-path scheduling: an effective technique for allocating task graphs to multiprocessors," *Parallel and Distributed Systems, IEEE Transactions on* , vol.7, no.5, pp.506-521, May 1996.

[20] Tilera, "Tile-GX3100", [Online]. Available: http://www.tilera.com/sites/default/files/productbriefs/TILE-Gx 3000 Series Brief.pdf.

[21] Standard Task Graph Set, "STG", [Online]. Available: http://www.kasahara.elec.waseda.ac.jp/schedule/

[22] I. Ahmad, Y.-K. Kwok, M.-Y. WU, and W. Shu, "CASCH: A Tool for Computer-Aided Scheduling," IEEE Concurrency, vol. 8, no. 4, pp. 21-33, October 2000.

[23] M.-Y. Wu, D. Gajski, "Hypertool: A Programming Aid for Message-Passing Systems," IEEE Transactions on Parallel and Distributed Systems, vol.1, no. 3, pp. 330-343, July 1990.

[24] Z. Wang and S. Ranka, "A Simple Thermal Model for MPSoC and its Application to Slack Allocation," *Proceeding of IEEE International Parallel & Distributed Processing Symposium*, 2010.

[25] W. Felter, K. Rajamani, T. Keller, and C. Rusu, "A performance-conserving approach for reducing peak power consumption in server systems," *In Proceedings of the 19th annual international conference on Supercomputing* (ICS '05). ACM, 2005.