Thermal-Aware, Power Efficient, and Makespan Realized Pareto Front for Cloud Scheduler

Saeeda Usman*, Kashif Bilal†, Nasir Ghani‡, Samee U. Khan*, and Laurence T. Yang∞
*Department of Electrical and Computer Engineering
North Dakota State University
Fargo, ND, USA, 58108-6050
Email: {saeeda.usman, samee.khan}@ndsu.edu

†Department of Computer Science
COMSATS Institute of Information Technology
Abbottabad, Pakistan, 22060
Email: kashifbilal@ciit.net.pk

‡Department of Electrical Engineering
University of South Florida
Tampa, FL, USA 33620-5399
Email: nghani@usf.edu

∞Department of Computer Science
St. Francis Xavier University
Antigonish, NS, Canada, B2G 2W5
Email: ltyang@stfx.ca

***Abstract*—The performance demands of modern computing applications have enormously accelerated the power density of on-chip devices. Consequently, not only the operational budget has increased exponentially, but also the temperature has experienced an alarming increase rate. The high chip temperatures pose serious threat to devices reliability. The aforementioned challenges necessitate the requirement of realizing efficient mapping methodologies to overcome resource exploitation. We present a convex optimization model that maximizes the power savings while satisfying the task completion deadline and maintaining chip temperature within acceptable bounds. Optimization is achieved by varying the frequency of operation of the core. The salient feature of the devised approach is the bounds of operation for the desired objectives. The acceptable operation limit manifests the elasticity of the computing resources in an autonomic manner to achieve a Pareto front for the optimized solutions. To validate our proposed method, extensive simulations are performed on a variable workload.**

***Index Terms*—Optimization, Pareto front, Performance, Temperature aware**

I. Introduction

The dynamic and promising services delivered by Cloud computing paradigm have strikingly elevated the demand of Cloud deployment (models). The paradigm orchestrates the computing resources, such as the processing cores, I/O resource, and storage to meet "on demand" client requirements. The aforementioned characteristic of Cloud has extensively scaled the service offering to leverage and productize functionality. However, to ensure that the agreed Service Level Agreement (SLA) is met, the Clouds needs to offer metering services to avoid resource exploitation.

To provide a single pane view of the resources status and achieve high levels of granular visibility, intelligent monitoring should by realized to track resource utilization. Due to the increase in chip power density, the offered computing resources are prone to predicaments, such as hardware failure, low reliability, and insecure multi-tendency. Indeed, task completion is the foremost priority of schedulers in Cloud. Nevertheless, thermal management and power consumption hold pivotal importance in achieving high-end functionality. Moreover, cost minimization can be accelerated by avoiding over-provisioning of the aforementioned resources.

Recently, a wide range of hardware and software based technique [2], [3], [8] have been proposed to control the power consumption of Chip Multi-Processors (CMPs). Although the management schemes could effectively reduce power depletion, they incur performance overhead in the form of thermal runaway. Motivated by this fact, the work presented in this paper address the abovementioned issue by considering the run-time information. Therefore, frequent monitoring of core temperature and operating frequency is required to lower the risk of chip overheating. We provide a methodology to mitigate the violation of peak power and temperature constraints, respectively.

To improve the performance of a scheduler in Cloud, we propose a temperature-aware power efficient methodology that judicially maximize performance and system reliability. The objective of this work is to optimize the cumulative performance of the resource allocation system. Intuitively, a convex optimization approach is devised to minimize the makespan, temperature, and power utilization of the scheduler. Our contribution circumvent the efficient management of power/temperature exploitation without comprising the task completion deadline.

Our major contributions are listed as follows:

- We develop a resource mapping heuristic that optimizes the performance using rigorous mathematical modeling. The scheduling decision space is constrained with a set of system specifications to attain the desired results.

- We model the problem to demonstrate the relationship between the frequency and the power consumption of the scheduling system in Cloud. The formulation unveils bounds on power and temperature utilization to dynamically adjust the resource utilization.
- The solutions that adheres to all of the constraints of power, makespan, and temperature constitute to the set of efficient or Pareto optimized solutions. Despite of the contradicting nature of the objectives, we perform efficient mapping of resources to fulfill the end user demands without the violation of any timing constraints.
- The chip temperature is kept into consideration while scaling the operating frequency to avoid cooling challenges and ensure safe operating temperature.
- The relevant Pareto front of high quality is obtained for the optimization of the three objectives. The power and temperature management is judicially performed using the proposed heuristic. Moreover, the deadlines of the tasks are preserved to ensure efficient performance.

The paper is organized as follows. Section II presents the system architecture. Section III provides the details of the system model followed by the preliminaries of the proposed model in Section IV. The problem formulation is presented in Section V. Section VI presents a discussion on the methodology adopted for the Pareto front approximation. Performance evaluation and simulation results are presented in Section VII. Section VIII discusses the related work, and Section IX concludes the paper.

II. System Architecture

Cloud service provider must ensure that the clients perpetually receive the Cloud services (resources) according to the agreed Quality of Services (QoS) level. Concurrently, the resources must be distributed efficiently and intelligently minimizing the resource wastage. Our goal is to optimally schedule the Cloud resources to fulfill the requirements. Allocation is mapped in a manner that the over provisioning of resources is prohibited. The scheduling mechanism detailed here promises to minimize the power consumption, temperature, and makespan of a scheduling within a Cloud environment. The objective is to optimize the task completion time such that the power conservation is not sacrificed while adhering to the temperature bound to overcome hot-spots. The scheduling mechanism is routed to generate a set of Pareto solutions, the *Pareto front*, to achieve the most efficient optimization for the Multi-Objective Problem (MOP) under consideration.

Definition (Pareto front). *A point $x^* \in X$ is called Pareto optimal if there is no $x \in X$ such that $F(x) < F(x^*)$. Then, $F(x^*)$ is said to be globally efficient. The set of all such optimal points contours the image (curve) called as the Pareto front.*

The goal of MOP performed in the work presented here is to identify the set of efficient points, $F(x_i)$ for $\forall\ x_i \in x^*$, that is able to represent the Pareto front, as shown in Fig. 1. The key concept is to find the desired optimal operating point that guarantees all the objectives without violating the set of constraints. Consequently, a set of Pareto efficient solutions, are acquired that characterize the improvement all of the objectives without worsening any.

III. System Model

Consider a scheduling system in Cloud. The scheduler performs the task allocation. The tasks are mapped on the set of machines that satisfy the task requirements.

A. Machines

The resource scheduler allocates the incoming tasks to a set of machines, $M = \{M_1, M_2, ..., M_k\}$. The machines are assumed to be equipped with a Dynamic Voltage/ Frequency Scaling (DVFS) module. A constant and negligible transition time between successive levels of the DVFS is assumed for the problem considered in this paper. Each machine of the set $M(M_j \in M)$ is characterized by the following attributes:

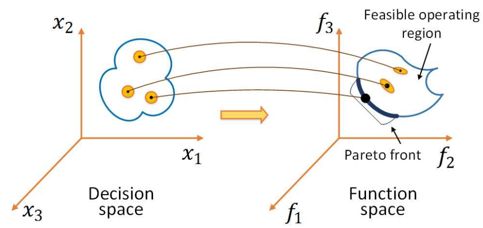


Fig. 1. Desired Pareto front for the set of efficient solutions

- The operations frequency of the machine, f_j, measured in hertz or cycles per unit time. By employing the DVFS, the frequency f_j can varied from f_j^{min} to f_j^{max}. The hierarchy of the frequency bounds is defined by the relation, $0 < f_j^{min} < f_j^{max}$. The frequency holds a linear relationship with the speed of the machine [6].
- The machine architecture, $A(M_j)$, that comprises of the storage specifications, speed rendered, and the kind of CPU utilized.

B. Tasks

Consider a metaset of tasks,$T = \{\tau_1, \tau_2, ..., \tau_n\}$. Each task, $\tau_i \in T$, is characterized by the following requirements:

- The time, t_i, required to complete the execution of the task. The Expected Time to Complete (ETC) is presumed to be known a priori.
- The machine architectural requirements, $A(\tau_i)$, that entails the task execution.
- The deadline, d_i, specifies the time at which the task execution must be performed. A successful mapping of tasks happen when all the constituting tasks of the set T are executed before the assigned deadlines.

IV. Preliminaries

In this section, we present the modeling basics of power and temperature management. As explained in the previous section, real-time tasks in a task set are supposed to execute on a set of machines. The machines used in the scheduling system are assumed to be equipped with the DVFS methodology. Therefore, each machine is enabled to switch between discrete levels of normalized frequencies, that is, $\{f_1, f_2, , f_L\}$. Where $0 < f_j^{min} = f_1 < f_2 < .... < f_L = f_j^{max}$.

A. Power Model

The power requirement is a cumulative sum of the idle and active mode expected power consumption. Such that:

$$\begin{aligned} P_{Total} &= P_{Dynamic} + P_{Static} + P_{Constant} \\ &= (\alpha . C_{EFF} . f . V^2) + (I_0 . V) + P_c, \end{aligned} \quad (1)$$

where $V, freq, C_{EFF}, I_0$, and α are the supply voltage, clock frequency, effective switch capacitance, leakage current and activity rate of the computing device, respectively. The dynamic power consumption refers to the power consumed in the active/dynamic mode. The static power consumption corresponds to the power dissipated regardless of switching activity. The term P_c relates to the power expended by various system component activities, such as memory/disk accesses.

The dynamic mode power reflects a quadratic relationship between the supply voltage and the power consumption of the system. Assuming a linear relationship between voltage and frequency, given as:

$$voltage \propto f,$$

The dynamic power dissipation becomes,

$$Power \propto Voltage^3,$$

or

$$Power \propto f^3.$$

The key design idea of DVFS is governed by the power-frequency proportionality relationship, such that a reduction in the clock frequency or supplied voltage, results in a cubic decrease in the power consumed. It is to be understood that the time to finish an operation is inversely proportional to the clock frequency. Such that:

$$time = 1/f. \tag{2}$$

Therefore, lowering the supply voltage also decreases the maximum achievable clock speed. Running the machine at a slower frequency can significantly reduce a computing devices' dynamic power consumption. On the contrary, reducing the frequency/voltage would substantially slow down the time to complete an operation. It is apparent from the equations listed above that one can reduce cubically the instantaneous power consumption, at the expense of linearly increased delay (reduced speed). Owing to this analysis, we adopt the DVFS-based frequency selection scheme to maximize the processor power savings.

B. Temperature Model

To model the temperature realization of a machine in a scheduling system, we follow the dynamic thermal model proposed by Skadron *et al.* [14] to characterize the thermal behavior of the processor. The model unifies the Resistance-Capacitance (RC) model and the temperature $(Temp)$ at a time instance t, such that:

$$Temp(t) = Temp_{stsd} \times (Temp_{stsd} - Temp_{start}) \times e^{-t/RC}, \tag{3}$$

where $Temp_{stsd}$ is the steady state temperature, $Temp_{start}$ is the initial temperature, R is the thermal resistance, and C is the thermal capacitance. The thermal resistance R and capacitance C are constants depending on the machine architecture. The steady state temperature of a task is the temperature that will be touched if vast number of occurrences of the task execute continually on the machine. It bears an almost linear relationship with the power consumed, and is given by:

$$Temp_{stsd} = (Power \times R) + Temp_{amb}, \tag{4}$$

where R is the thermal resistance, as explained earlier and $Temp_{amb}$ is the ambient temperature of the machine/core. The power consumption of tasks differ significantly depending on the nature of the task. Therefore, we can say that the steady state temperatures of tasks in a task set are different. In the quest to improve performance, continuous scaling of supplied voltage has been the focal point. Consequently, high operational frequency is exercised to meet high power needs. The dynamic power remains unaffected with the change in temperature. Nevertheless, the static power loss increases exponentially with temperature. The leakage current, I_0, is given as:

$$I_0 = I_s(ATemp^2 e^{(\mu_1 V + \mu_2)/Temp} + Be^{(\mu_3 V + \mu_4)}), \tag{5}$$

where I_s, V, and $Temp$ is the initial leakage current, voltage supplied, and the operating temperature, respectively. Whereas, $A, B, \mu_1, \mu_2, \mu_3, and\ \mu_4$ are constants with values determined empirically. The leakage current thereby increases exponentially with temperature, as shown in Fig. 2.

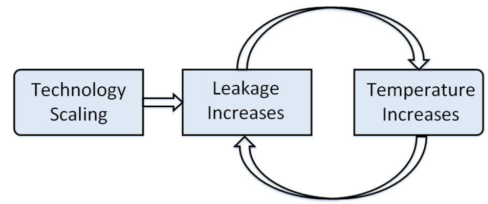


Fig. 2. Impact of technology scaling

Therefore, to avoid hotspots and thermal runaway temperature effects cannot be neglected. The total power consumption of a task running on a machine is given by;

$$P_{total} = C_{EFF} f^3 + C_1 freq + C_2 Temp_f, \tag{6}$$

where C_1 and C_2 are the curve fitting constants [5]. Based on the above mentioned mathematical model to optimize the power consumption of a machine we adopt the DVFS-based power scaling. The temperature control is gauged by an optimum temperature bound check given in the next section.

V. Problem Formulation

Consider a given a resource allocation system that comprises a set of machines, M, and a set of tasks, T. The scheduler is required to map tasks on the machine set, such that all the characteristics of the tasks and the deadline constraint of T are fulfilled. We term this assignment as a feasible task to machine mapping. A feasible task to machine mapping happens when each task $\tau_i \in T$ can be mapped to at least one M_j subject to all of the constraints associated with each task, such that the computational time, architecture, and deadline. The aforementioned requirements of the tasks are recorded as a Boolean operator (x_{ij}).

The task to machine mapping is performed such that, a minimization of the cumulative instantaneous power (P_{ij}) consumed by the machines in the scheduling system, the temperature and the makespan of the set of tasks, MS_{ij},is ensured. Power management is achieved by regulating the voltage and frequency supplied using the DVFS. The DVFS methodology exploits the convex relation between the power expended by a machine to the voltage and frequency exploited. The motivation of using the DVFS technique is to expand the task execution time using frequency and voltage reduction to minimize the overall power consumption. Table I presents the legend explanation.

Objective function

$$\begin{aligned} O &= x_{ij}[Min(MS_{ij} + P_{ij} + Temp_{ij})] \\ &= \left(\sum_i \sum_j MS_{ij} x_{ij}\right) \alpha_{MS} + \left(\sum_i \sum_j P_{ij} x_{ij}\right) \alpha_p \\ &\quad + \left(\sum_i \sum_j temp_{ij} x_{ij}\right) \alpha_{temp}. \end{aligned} \tag{7}$$

s.t. $\forall i, j, where (i > 0 and j > 0)$.

Bounding weight parameter

$$\sum_i \sum_j \alpha_{ij} \leqslant 1, \tag{8}$$

where α= proportional weight parameter

$$\sum_i \sum_j x_{ij} = 1, \qquad \forall j \in M \quad and \quad \forall i \in T \tag{9}$$

$x_{ij} = 1\ if\ task\ "i" is\ allocated\ to\ node\ "j"\ otherwise\ 0$

TABLE I
LINEAR PROGRAM TERMINOLOGY

Variable	Description
i	Task indication variable/subscript
j	Indicates node/machine
x_{ij}	Binary variable
MS_{ij}	Makespan (completion time of task set i on machine j)
t_i	Time to complete task i
P_{ij}	Power consumption of machine j when task i is performed
Ψ_{ij}	Time of execution of task i on machine j
f	Frequency of the core/node
S	Storage memory required
F	Finishing time
C	Job consolidation resources
τ_i	Task $i \in T$
M	Set of machines
$Temp$	Temperature of machine
d_i	Deadline of the task i
T	Set of tasks to be performed

Bounding Power Consumption (A constraint)

$$\sum_i \sum_j p_{ij} \leqslant P_j^{max} \qquad \{P_j > 0\}. \tag{10}$$

Set of Constraints:

(a) Execution time of task i on node j, $\Psi_{ij} \leqslant deadline_i$ $\forall_i = tasks$
(b) Frequency of node, $f_j^{min} \leqslant f_{ij} < f_j^{max}$ $(0 < f_j^{min})$
(c) Resource Consolidation, $C_{ij} \leqslant C_j$
(d) Storage capacity, $S_{ij} \leqslant S_j$
(e) Finish time of machine, $F_j \leqslant MS_{maximum}$
(f) $temp_{min} \leqslant temp_j \leqslant temp_{max}$

Assumptions in the formulation:

- The task characteristics are not expected to change during the execution course. That is, the deadline of task completion and resource requirement remains the same.
- The expected execution time of a task is considered as a decision criteria for scheduling a task on a node.
- P_{ij}, represents the power consumption of all of the components of a machine.
- A task in execution process is not stopped until completion. However, the assigned resources may change, e.g. power consumption of the node.
- The cooling power of the scheduler is not included in calculations.

VI. PARETO FRONT APPROXIMATION

We focus on the optimization problem with three objective functions, $f_1(x), f_2(x)$, and $f_3(x)$. Although, there are variety of computational methods for procuring the aforementioned objective. The methodology incorporated for the approximation of the Pareto front used in our work is the dual simplex procedure. Each objective function is assigned a certain weight and the point of optimization is adaptively determined. At each iteration the non-dominated points are identified to construct the set of Pareto optimal points.

Fig. 3 depicts the linear programming model considered in our paper. The co-ordinates represent the independent variables, depicted by the objective functions. The constraints restricts the allowable feasible region to a specified operating area, as shown in the Fig. 3. The dual simplex methodology will enable the system to operate in the optimal region. The desired region is identified by imposing the constraints and bounding the objective function by the feasible vectors. The feasible solutions that are dominated are discarded in the quest to find solution that after better optimization to make the set of non-dominated solutions. The Pareto front approximation seeked in this work, is given as:

$$\forall k \qquad f_k(x^*) \leqslant f_k(x), \tag{11}$$

$$\exists k \qquad f_k(x^*) < f_k(x). \tag{12}$$

where $f_k(.)$ represents the k^{Th} objective function. Nevertheless, x^* depicts the non-dominated solutions and x constitute the dominated solution. The elements of the set $f_k(x*)$ indicates the desired optimized solution set and compose the coveted Pareto front.

Algorithm 1 Round to the nearest integer solution while maintaining the constraints

INPUT: The number of tasks, τ to perform, the number of machine, M, and objective to minimize;
OUTPUT: Optimal solution for the problem by executing the benchmark of required performance level;
INITIALIZATION: The control parameters of the objective function are provided;

Step 1
if $f_1, f_2, ..., f_n \ \geqslant \ 0$ **then**
  L_P problem is optimal
else
  choose the most negative $f_1 < 0$
Step 2
  compute the pivot element

$$\min_{1 \leqslant k \leqslant m} \{b_k/a_{kl} \mid a_{kl} > 0\}$$

  obtain a basic solution and update the objective function row as:
Step 3
  $f_N = f_N - \beta(\gamma_p)$
  where $\beta = b_\gamma / a_{\gamma l}$
  and γ_p= pivotal row element
  update and generate the new simplex tableau
end if
Go to Step 1

A. Dual Simplex Method for the Linear Programming

We assume that the mapping (x_{ij}) is only fulfilled when the resource consolidation/architectural constraints of the tasks are satisfied. The resource consolidation refers to the desired level of storage, memory, power, and Virtual Machines (VMs) required to perform a particular task or set of task on a machine. The constraint optimization problem is resolved with the simplex method of linear programming. The feasible intersection points are enumerated using the complex (Simplex) method. Nevertheless, the worst point will be replaced by a new and better point using the aforementioned methodology. A variation in the parameter f is used to achieve an optimized solution to the problem.

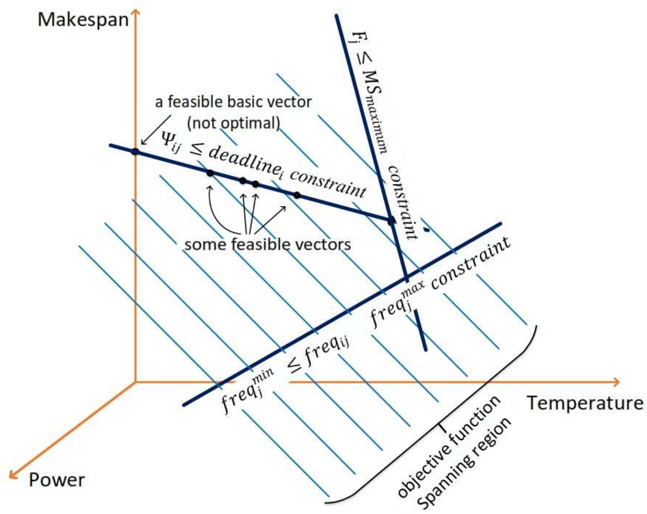


Fig. 3. Linear programming model

The systems' modeling is initiated by normalizing the constraints. The inequalities in the constraints list are converted into equations by adding a slack, and surplus respectively, such as:

$$Constraint_i + slack_i = feasible\ limit/bound.$$

The simplex tableau for the objective function is generated comprising of the objectives, constraints, and artificial variables. The elements of the objective function play a pivotal role in the optimization methodology. The simplex method operations work in the form of a tableau. For every iteration a new tableau is generated to indicate the convergence. Moreover, the new tableau highlights the objectives function values that needs to optimized to achieve the overall optimization goal. The optimization process is characterized by the evaluation of feasible intersection points. Table II represents a generalized simplex tableau.

In the Table II, the most negative co-efficient in the objective function row determines the pivot column. The columns pertaining to the variables S_i simply record the slack and surplus term of each of the constraint. First, the simplex procedure is employed to find the pivotal element. The pivot identifies the next intersection point to be evaluated that improves the objective. Thereafter, Gaussian elimination step is followed to attain the next simplex tableau.

The entries of the objective function row in the Table II determines the decision of generation a new tableau. The iterations stop until every element of the objective function takes a non-negative value, while adhering to the bounds of operation. Algorithm 1 details the procedure to achieve the non-dominated solution for the linear convex problem. A weighted sum approach is employed to generate the Pareto front. The admissible feasible solutions are generated. The solution that form the desired convex combination of the objectives are retained and vice versa. Formally, the vertices that restrict the objective function to the optimal corner point, make the set required basic solution. The algorithm initiates with the test of optimality that checks the current state of objective parameters. That is, if the elements constituting the last row, are positive, the optimality condition is already reached. Otherwise, the procedure evaluates the identification of the pivotal element that is triggered by the most negative entry of the simplex tableau. Using the elimination methodology the new transformed tableau is generated. The same check and do procedure is followed until the optimality is reached.

VII. Simulation Results

To validate our proposed methodology, the scheduling operations performed are implemented in Matlab. To evaluate the proposed algorithm, we used a core i-7 desktop PC with 3.4 GHz of CPU speed and 8 GB of RAM. The dimensions of the tasks executed are as large as 10,000 tasks by a total of 20 computing nodes. The task mapping is restricted to the constraints and operational bounds listed in Section IV. The objective of the simulations performed is to maximize the performance while minimizing the power and temperature factor. The algorithm 1 depicts the weight parameter, α, for each of the desired objective. The purpose of utilizing α is to define the importance of a specific objective in a controllable manner.

Performance is quantified by the completion of a task by a machine in the targeted deadline. Note that, if the task completion is performed in the range of allowable power and temperature constraints, it is specified as efficient mapping. The solutions that adhere to the system specifications of defined objectives and constraints are used to construct the coveted Pareto front. Fig. 4 depicts the ability of the proposed algorithm to efficiently minimize the desired objectives.

To validate the effectiveness of proposed technique, we perform a comparison of the proposed methodology with LP and greedy heuristics. A set of five machines is used to test the compliance to temperature and power bound. The maximum power budget for each node is 250W for a makespan time range of upto 3000 seconds. The peak temperature is constrained at 85°C. Figure 5 (a-f) shows the response of each of the five nodes working under the abovementioned scheduling methodologies. Figure 5 (a-c) represents the node response to the allowable temperature bound under the LP, greedy, and the proposed technique, respectively. However, Fig. 5 (d-f) shows the adherence of the nodes to the allowable power budget under the aforementioned methodologies/procedures. The results obtained reveals that the LP and greedy approaches violates the bounds on the maximum temperature and power consumption per node. However, the proposed methodology is efficient in adhering to the system constraints of power and temperature.

In Figure 4, the non-dominating solutions assume a computationally calculated Pareto front for the optimized solutions. However, the sub-optimal solutions are represented as dominating solutions. The non-dominating solution set sweeps out the dominated solutions from the knee region of the curve. The solutions that are less optimized form the tails of the Pareto-optimized graph. Neverthe-

TABLE II
GENERALIZED SIMPLEX TABLEAU

Iteration	P_{ij}	f_{ij}	Ψ_{ij}	$Temp_{ij}$	F_j	S_1	S_2	S_3	S_4	S_5	S_6	B*
						1						
	Values should be obtained as per the task to machine assignment.						1					
								1				
									1			
										1		
											1	
	Values of the objective function											

**Values corresponding to the maximum limit (Boundary or Extreme)*

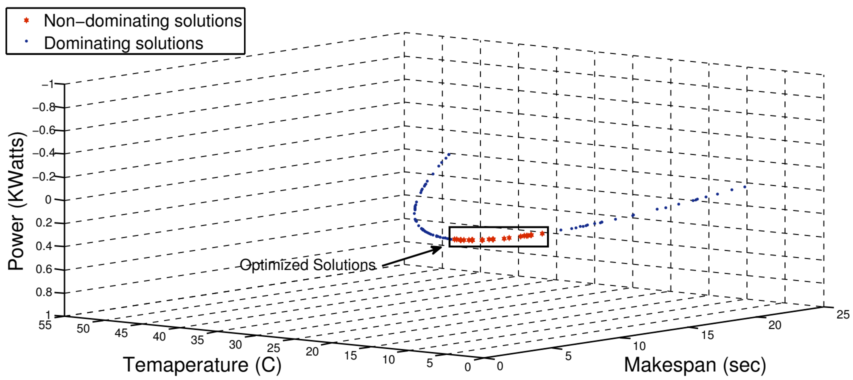
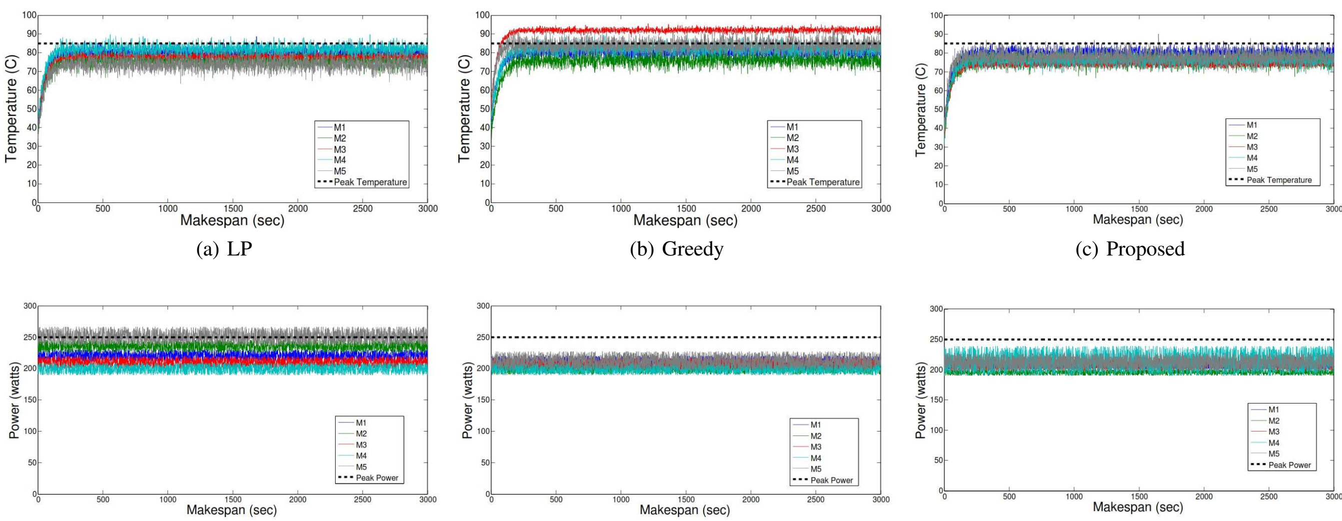


Fig. 4. Pareto front of the optimized solution set



(a) LP (b) Greedy (c) Proposed

(d) LP (e) Greedy (f) Proposed

Fig. 5. Comparison of proposed methodology with LP and greedy heuristics

less, the solutions that comprise the knee region are comparatively better in minimizing all of the three objectives. Consequently, the overall performance of a scheduler is improved by adhering to the abovementioned details. The allocated tasks when are performed by the available set of machines following the methodology depicted in the paper, a Pareto optimized front curve may be obtained.

VIII. Related Work

A large number of hardware and software techniques, for example [3], [9], [12] and [7] have proposed by researchers to improve the energy profile of multi-core systems. The traditional power saving strategies focus on scaling the voltage and frequency of the core to meet the allowable power level. However, temperature received less attention. Consequently, reliability and decrease in the life-time of the chip resulted as a trade-off. Therefore, researchers over the last decade, emphasize the need of *Dynamic Thermal Management (DTM)* [14] and [11] for safe chip operation and to reduced cooling cost.

The work presented by authors in [10], [13], [16] perform optimization of power consumption while guaranteeing the required performance. Nevertheless, the aforementioned methodologies optimizes the power and performance, but during the optimization. Authors in [17] speculates the chip thermal management requirement and devised methodologies to attain the chip temperature optimization. In Ukhov *et. al* [15] the authors propose a Steady State Dynamic Temperature Profile (SSTDP) to realize temperature-aware reliability model. The technique consider mitigating the thermal cycling failure. However, transient faults and their management is not catered. Moreover, power optimization is not entertained while achieving reliability.

To circumvent the difficulty to manage the thermal and power minimization, a large body of research has been done. Fisher *et. al* [4] employed optimal speed scheduling of cores for thermal and power optimization. The energy consumption is minimized based on the condition that the multi-core thermal model is present at the design time. Nevertheless, despite of the prediction of thermal behavior a slight variation in run-time specifications can cause serious problems in achieving the estimated level of optimization. Minimum number of active cores or server consolidation [1] approach is used to mitigate the power loss in multi-core environment. Although, the aforesaid approaches try to minimize the power budget, they lack an accurate mathematical formulation and a rigorous algorithmic solution to solve the optimization problem. A uniform task/job profile is not an efficient solution to minimize the power and thermal dissipation. Therefore, we present a dynamic approach that depending on the task requirements adjust the assigned frequency to meet the system power and temperature budget.

Significantly, different from the above listed work, this study explore the scheduling decision space to optimize the performance of multi-core system. The temperature and power utilization is capped and dynamically adjusted while meeting the performance requirement of the system.

IX. Conclusions and Future Work

The exponential increase in the demand of Cloud deployment is constrained by the prohibitively high operational cost. To address the abovementioned issue, the work presented in our paper combined the benefit of power- and temperature-awareness in multi-core systems. A coherent framework of power, temperature, and makespan optimization is proposed to attain promising performance. Using the frequency of operation as selection criteria the task allocation is mapped to simultaneously minimize the aforementioned objective function entities. We proposed a formulation that caters the heterogeneity among resources and proposes bounds of operation to adjust to the varying demand of power, frequency, and temperature. Moreover, to define precedence a weighted approach is utilized to significantly impact the desired objective and guarantee desired results.

The results reveal that the algorithm proposed is efficient in obtaining the trade-off front and removing the dominated solutions. The trade-off comprises the Pareto front and comprises of the non-dominated solutions. For future work, we plan to investigate the methodology on an extended scale of performance objectives. The particular domains of interest encompass throughput maximization and reduction of network congestion.

Acknowledgments

The authors are grateful to Indranil Sengupta, Muhammad Awais, Muhammad Jawad, and Nikita Barabanov for the valuable suggestions, reviews, and guidance.

References

[1] [Online]. Available: http://www.sun.com/x64/intel/consolidate-using-quadcore.pdf

[2] A. Abbas, M. Ali, A. Fayyaz, A. Ghosh, A. Kalra, S. U. Khan, M. U. S. Khan, T. De Menezes, S. Pattanayak, A. Sanyal *et al.*, "A survey on energy-efficient methodologies and architectures of network-on-chip," *Computers & Electrical Engineering*, vol. 40, no. 8, pp. 333–347, 2014.

[3] K. Bilal, A. Fayyaz, S. U. Khan, and S. Usman, "Power-aware resource allocation in computer clusters using dynamic threshold voltage scaling and dynamic voltage scaling: comparison and analysis," *Cluster Computing*, pp. 1–24, 2015.

[4] N. Fisher, J.-J. Chen, S. Wang, and L. Thiele, "Thermal-aware global real-time scheduling on multicore systems," in *15th IEEE Real-Time and Embedded Technology and Applications Symposium,*, 2009, pp. 131–140.

[5] H. Huang, V. Chaturvedi, G. Quan, J. Fan, and M. Qiu, "Throughput maximization for periodic real-time systems under the maximal temperature constraint," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 13, no. 2s, p. 70, 2014.

[6] S. U. Khan and I. Ahmad, "A cooperative game theoretical technique for joint optimization of energy consumption and response time in computational grids," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 3, pp. 346–360, 2009.

[7] D. Kliazovich, P. Bouvry, and S. U. Khan, "Dens: data center energy-efficient network-aware scheduling," *Cluster computing*, vol. 16, no. 1, pp. 65–75, 2013.

[8] J. Kołodziej, S. U. Khan, L. Wang, and A. Y. Zomaya, "Energy efficient genetic-based schedulers in computational grids," *Concurrency and Computation: Practice and Experience*, 2012.

[9] P. Lindberg, J. Leingang, D. Lysaker, K. Bilal, S. U. Khan, P. Bouvry, N. Ghani, N. Min-Allah, and J. Li, "Comparison and analysis of greedy energy-efficient scheduling algorithms for computational grids," *Energy Aware Distributed Computing Systems, John Wiley & Sons, Hoboken, NJ, USA*, 2012.

[10] P. Lindberg, J. Leingang, D. Lysaker, S. U. Khan, and J. Li, "Comparison and analysis of eight scheduling heuristics for the optimization of energy consumption and makespan in large-scale distributed systems," *The Journal of Supercomputing*, vol. 59, no. 1, pp. 323–360, 2012.

[11] S. Murali, A. Mutapcic, D. Atienza, R. Gupta, S. Boyd, L. Benini, and G. De Micheli, "Temperature control of high-performance multi-core platforms using convex optimization," in *Design, Automation and Test in Europe, 2008. DATE'08*, 2008, pp. 110–115.

[12] J. Shuja, K. Bilal, S. A. Madani, and S. U. Khan, "Data center energy efficient resource scheduling," *Cluster Computing*, vol. 17, no. 4, pp. 1265–1277, 2014.

[13] J. SiYuan, "A novel energy efficient algorithm for cloud resource management," *International Journal of Knowledge and Language Processing*, vol. 4, no. 2, 2013.

[14] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Trans. Archit. Code Optim.*, vol. 1, no. 1, pp. 94–125, Mar. 2004.

[15] I. Ukhov, M. Bao, P. Eles, and Z. Peng, "Steady-state dynamic temperature analysis and reliability optimization for embedded multiprocessor systems," in *Proceedings of the 49th Annual Design Automation Conference*, 2012, pp. 197–204.

[16] L. Wang, S. U. Khan, D. Chen, J. Kołodziej, R. Ranjan, C.-z. Xu, and A. Zomaya, "Energy-aware parallel task scheduling in a cluster," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1661–1670, 2013.

[17] J. Zhou and T. Wei, "Stochastic thermal-aware real-time task scheduling with considerations of soft errors," *Journal of Systems and Software*, 2015.