Automated Techniques for Energy Efficient Scheduling on Homogeneous and Heterogeneous Chip Multi-processor Architectures

Sushu Zhang Karam S. Chatha
Department of Computer Science and Engineering, Arizona State University
Tempe, Arizona, USA 85287
Email: sushu.zhang@asu.edu, kchatha@asu.edu

Abstract— We address performance maximization of independent task sets under energy constraint on chip multi-processor (CMP) architectures that support multiple voltage/frequency operating states for each core. We prove that the problem is strongly NP-hard. We propose polynomial time 2-approximation algorithms for homogeneous and heterogeneous CMPs. To the best of our knowledge, our techniques offer the tightest bounds for energy constrained design on CMP architectures. Experimental results demonstrate that our techniques are effective and efficient under various workloads on several CMP architectures.

I. Introduction

In this decade, computer architecture has entered a new "multi-core" era with the advent of Chip Multiprocessors (CMPs). Many leading companies, Intel, AMD and IBM, have successfully released their multi-core processor series, such as Intel IXP network processors [11], the Cell processor [7], the AMD Opteron^{TM} etc. CMPs have evolved largely due to the increased power consumption in nanoscale technologies which have forced the designers to seek alternative measures instead of device scaling to improve performance. Increasing parallelism with multiple cores is an effective strategy. However, the power dissipation challenges do not disappear in the CMP regime. In fact, the power optimization problem in CMPs is quite complex as these architectures include multiple heterogeneous processing cores. Even though there exists a large body of work on power optimization in uni-processor architectures, there is still little understanding of the power-performance challenges on CMPs [15, 13].

Dynamic voltage and frequency scaling (DVFS) exploits the cubic relationship between power consumption and supply voltage to minimize power at the expense of linear slow down in operating frequency. Several current CMPs support voltage/frequency (v/f) scaling options for individual processing cores. In the Cell processor, each SPE can work at 5 supply voltages ranging from 0.9V to 1.3V [7]. Therefore, the application developer can develop a core-level DVFS policy to maximize performance within a given energy budget. However, the problem is quite complex as it also includes the determination of the mapping between the tasks and processing elements.

In this paper, we propose polynomial time off-line approximation algorithms for the energy constrained scheduling problem on homogeneous/heterogeneous CMP architectures that support core-level DVFS. The proposed techniques jointly address two key problems for energy efficient application development on CMP architectures: 1) the mapping of tasks to processing elements (PE) and 2) selection of discrete v/f state for execution of each task. The objective of the techniques is to maximize the performance of an application subject to an energy budget. We prove that the energy-efficient mapping and scheduling (EMMS) problem as described is strongly NP-hard. We then propose polynomial time techniques for homogeneous and heterogeneous CMP architectures that can be shown to generate solutions whose performance (latency or makespan) is no more than twice (2-approximation) of the optimal. To the best of our knowledge the proposed techniques offer the tightest quality bounds for the EMMS problem. Our experimentation results demonstrate that for practical instances of the problem the performance of our solutions is on an average no greater than 1.43 of the optimal.

The paper is organized as follows: Section II discusses the previous work, Section III defines the problem and proves that it is strongly NP-hard, Section IV presents the approximation algorithms for the homogeneous and heterogeneous CMPs, Section V presents the experimental results, and finally Section VI concludes the paper.

II. Previous Work

The existing techniques for energy-efficient scheduling on CMPs can be classified into several categories based on different metrics: i) the laptop problem [17, 5, 6, 9, 13] versus the server problem [19, 4, 15] ii) continuous [17, 5, 6] versus discrete v/f states [15, 13, 9, 19] iii) heuristic [13, 15, 19, 4] versus approximation [17, 5, 6, 9] techniques.

Bunde et al. [5] classified the energy efficient scheduling problems into the laptop problem and the server problem. The former fixed the energy consumption to maximize schedule performance, while the latter fixed the schedule performance to minimize energy consumption. Jha et al. [14] introduced different variations of the both problems with more considerations such as the task models [17, 6], the communication links [19, 3] and the synthesis costs [9]. Our work belongs to the laptop problem, which asks "given an energy budget, what is the best schedule to maximize performance". We focus on independent non-preemptable task set and assume all the tasks arrive at the same time instance.

We focus on the approximation techniques for the problem that can generate solutions with guaranteed quality bounds. The existing heuristic techniques [15, 13, 19, 4] cannot satisfy this property. Pruhs et al. [17] proposed a polynomial time

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

approximation scheme based on load balancing for the energy-efficient scheduling problem. Bunde [5] extended the work by Pruhs et al. and gave an exact algorithm for multiprocessor makespan minimization of equal-workload jobs. Chen et al. [6] summarized their approximation techniques on several variants of the energy-efficient scheduling problem. However, all of these techniques assumed that v/f could be scaled continuously. As we know, most commercial processors only support discrete v/f states and the optimal v/f as generated by the previous techniques may not be available. In the discrete v/f domain, Andrei et al. [3] presented a MILP formulation with multiple considerations for the energy-efficient scheduling problem. Hsu et al. [9] considered an independent task set with EDF/RM schedule and provided an (m+2)-approximation algorithm to minimize the allocation cost within an energy budget. In contrast, we propose 2-approximation polynomial time techniques for the EMMS problem on homogeneous and heterogeneous CMPs. To the best of our knowledge, our techniques offer the tightest quality bounds for the EMMS problem.

III. Problem Description

Consider a CMP composed of m PEs denoted by the set $\Phi = \{pe_1, \ldots, pe_i, \ldots, pe_m\}$. Each PE consists of a DVFS equipped processor, a local memory and a globally coherent DMA engine. An interconnect bus is provided for the communication between the PEs. On each pe_i, there is an available active voltage state set $\Psi_i = \{s_1, \ldots, s_k, \ldots, s_{l_i}\}(|\Psi_i| = l_i)$. We assume the local memory is large enough to hold all the tasks.

The energy-efficient multiprocessor mapping and scheduling (EMMS) problem is described as follows.

Given a target multiprocessor chip CMP, n independent non-preemptable tasks $\Gamma = \{\tau_1, \ldots, \tau_j, \ldots, \tau_n\}$ to be executed on the CMP, the objective is to maximize the chip-level throughput such that each task is scheduled at a unique v/f state on one of the PEs, and the total energy consumption is no more than an energy budget C.

We assume for each task τ_j, c_{ijk} and t_{ijk} are given as the energy consumption and the worst case execution time (WCET) of the task on $pe_i \in \Phi$ at v/f state $s_k \in \Psi_i$, respectively. And all the tasks arrive the CMP at time zero. The objective to maximize the chip-level throughput can be transformed to minimize the overall completion time (makespan) of the task set. In this paper, we focus on off-line provable approximation techniques for the EMMS problem. The Integer Linear Programming (ILP) formulation of the EMMS problem, named $\mathbb{P}1$, is as follows:

$$\min \mathrm{T}$$

$$s.t. \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l_i} c_{ijk}x_{ijk} \le C \quad (1a)$$

$$\sum_{j=1}^{n}\sum_{k=1}^{l_i} t_{ijk}x_{ijk} \le T, \forall pe_i \in \Phi; \quad (1b)$$

$$\sum_{i=1}^{m}\sum_{k=1}^{l_i} x_{ijk} = 1, \forall \tau_j \in \Gamma; \quad (1c)$$

$$x_{ijk} = \{0,1\}, \forall pe_i \in \Phi, \forall \tau_j \in \Gamma, \forall s_k \in \Psi_i. \quad (1d)$$

Here x_{ijk} is 1 if and only if τ_j is executed at v/f state s_k of the pe_i, otherwise 0. Constraint (1a) specifies that the total energy consumption is no more than C. Constraint (1b) describes that the overall throughput is limited by the completion time of tasks on each PE. Constraint (1c) ensures that each task is executed on one voltage of some PE.

Theorem 1. *The EMMS problem is strongly NP-hard.*

Proof. We prove the strongly NP-hardness by showing that a well-known strongly NP-hard problem, the minimum makespan scheduling (MMS) problem [8], is a special case of the EMMS problem. When the processing time of each task is fixed and there is no energy budget constraint, the EMMS problem becomes a MMS problem with an arbitrary m. □

Hochbaum et al. [8] discusses several research results on approximation algorithms for the MMS problem. However, those results are for the classical MMS problems without consideration of energy budget or v/f states. In this work, we focus on approximation techniques for the EMMS problem. Standing on the shoulders of giants, we extend some useful ideas for the MMS problems to address the EMMS problem. In the following section, we present a 2-approximation algorithm for scheduling on homogeneous CMP by extending the LP rounding method for the MMS problem with identical machines [8]. Then, we propose a 2-approximation algorithm for scheduling on heterogeneous CMP based on the solution for the MMS problem with unrelated machines (the generalized assignment problem) [8, 18]. In contrast to the original algorithms [8, 18] for the MMS problems, both of our algorithms can deal with simultaneous v/f state assignment and task to PE mapping. In this work, the WCET of tasks is assumed to be integral as the cycle numbers in cores, and the switching overhead between v/f states is negligible.

IV. Approximation algorithms

In this section, we propose polynomial time approximation algorithms for the EMMS problem. Initially, a tight lower bound of the optimal EMMS is achieved by a binary search, and then the scheduling algorithms for the homogeneous CMP and heterogeneous CMP are proposed based on a fractional schedule. Both of the scheduling techniques are justified to be 2-approximation algorithms of the optimal EMMS.

A. Finding a tight lower bound of the optimal

In general, the LP relaxation of an ILP problem is an effective way to obtain the lower bound of the optimal. However, sometimes the LP relaxation result is not a tight lower bound. Consider the LP relaxation of $\mathbb{P}1$ by replacing $x_{ijk} \in \{0,1\}$ with $x_{ijk} \ge 0$, denoted as $\mathbb{P}1\mathbb{L}\mathbb{P}$. Suppose that we have two identical PEs, one single task, and each PE is only equipped with one v/f level. Assume the WCET of this task is t on the PE. The optimal makespan of $\mathbb{P}1$, denoted as $\mathrm{T}^\star$, would be t. However, the naive LP relaxation gives a solution where the task is split into equal halves on the two PEs. The optimal makespan of the LP relaxation is $\frac{1}{2}t$. The bound on $\mathrm{T}^\star$ is not tight because the WCET of the single job is larger than the lower bound. To avoid this case and achieve a tighter lower bound of $\mathrm{T}^\star$, we include a property of the optimal solution of

P1LP-OPT:
$l = T_{LB}, r = T_{UB}$
while $(l < r)$
$\{ h = \lfloor \frac{l+r}{2} \rfloor$.
if $(probe(h) = success)$ then $r = h$;
else $l = h$;$\}$
return $T^{\star}_{\mathbb{P}1\mathbb{L}\mathbb{P}} = r$ and S;

$probe(T)$:
Let $\mathrm{T}_d = \mathrm{T}$, solve $\mathbb{P}2\mathbb{L}\mathbb{P}$ by the simplex method;
if $(C_s \leq C)$) return $success$ and the solution S;
else return $failure$;

Fig. 1. An optimal algorithm for $\mathbb{P}1\mathbb{L}\mathbb{P}$

the ILP as an extra constraint. Thus, this constraint would not affect $\mathrm{T}^{\star}$.

$$\text{if } t_{ijk} > T, x_{ijk} = 0; \tag{2}$$

Since the if-then constraint is not easy to be linearized because of the unknown T, we introduce another problem, $\mathbb{P}2$, which includes this constraint. $\mathbb{P}2$ is described as "given a deadline T_d for the task set, what is the best schedule with minimum energy consumption". The ILP formulation is as follows:

$$\min C_s = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{l_i} c_{ijk} x_{ijk}$$

$s.t.$ Constraint (1b)(1c)(1d)(2) and replace T by T_d in (1b)(2).

In $\mathbb{P}2$, the if-then constraint can be transformed to a preprocessing step by setting values of some x_{ijk}, since T_d is given.

Let the $T^{\star}_{\mathbb{P}1\mathbb{L}\mathbb{P}}$ be the optimal makespan of the $\mathbb{P}1\mathbb{L}\mathbb{P}$ problem with Constraint (2). Based on the linear relaxation of $\mathbb{P}2$, named as $\mathbb{P}2\mathbb{L}\mathbb{P}$, $T^{\star}_{\mathbb{P}1\mathbb{L}\mathbb{P}}$ is found by the P1LP-OPT algorithm in Figure 1. In P1LP-OPT, the T_{LB} is set as $\min\{t_{ijk}\}$ and the T_{UB} is set as $n \cdot \max\{t_{ijk}\}$, $\forall pe_i \in \Phi, \tau_j \in \Gamma, \forall s_k \in \Psi_i$. Then, we have the following lemma. The proof is omitted here since it is similar to that for the MMS problem [18].

Lemma 1. *The binary search based on $\mathbb{P}2\mathbb{L}\mathbb{P}$ in the P1LP-OPT algorithm finds the optimal solution $T^{\star}_{\mathbb{P}1\mathbb{L}\mathbb{P}}$ of $\mathbb{P}1\mathbb{L}\mathbb{P}$.*

The P1LP-OPT returns an optimal fractional schedule S. For each $x_{ijk} > 0$ in S, $t_{ijk} \leq T^{\star}_{\mathbb{P}1\mathbb{L}\mathbb{P}}$, because of Constraint (2). In the following subsections, we present the scheduling techniques based on S for the homogeneous CMP and the heterogeneous CMP.

B. Scheduling on Homogeneous CMP

Homogeneous (or symmetric) CMP consists of m identical PEs. The scheduling problem on homogeneous CMP is easier than that on heterogeneous one, because the active v/f state space is independent of the PEs in the CMP. In other words, a task requires the same amount of WCET and consumes the same energy on a particular active state among all the PEs. Based on this property, we propose a simple 2-approximation technique in Figure 2.

Observe that the linear relaxation of $\mathbb{P}2$ after the preprocessing step for Constraint (2) includes at most $m + n$ constraints in addition to the non-negativity conditions. Therefore, each basic solution has at most $m + n$ basic variables which may take positive values while the other non-basic variables take the value zero. The simplex method searches among the basic solutions and generates an optimal solution of this form [16]. Thus, if $n > m$, there are at most m tasks that get split. Based on the property, we have

P_{sym}:
Step 1: Achieve the fractional schedule S from P1LP-OPT;
Step 2: For each τ_j, select the s_k with the associated smallest c_{ijk} with positive x_{ijk} in S;
Step 3: If $n \leq m$, schedule the tasks to disjoint PEs; else:
(a) schedule the tasks with the integral x_{ijk} from S;
(b) schedule the remaining tasks with the determined s_k to arbitrary disjoint PEs.

Fig. 2. A 2-approximation for EMMS problem with homogeneous CMP

Theorem 2. *The makespan of the schedule from the P_{sym} algorithm is at most twice of the optimal.*

Proof. In Step 1, the P_{sym} algorithm computes a fractional assignment from P1LP-OPT. After Step 2, the s_k with the smallest c_{ijk} for each τ_j is selected, when the t_{ijk} is associated with a positive x_{ijk}. The energy budget constraint is satisfied. At Step 3, we consider the two cases here.

- Case $n \leq m$: It is clear the n tasks can be assigned to disjoint m PEs. Thus, the overall makespan is determined by the WCET of single task on each PE. Because the schedule S satisfies Constraint (2), the makespan is $\leq T^{\star}$. Because $T^{\star}$ is the optimal, the result from P_{sym} is the optimal.
- Case $n > m$: After Part (a) in Step 3 of the P_{sym}, because of constraint (1b), the maximum completion time of this part is no more than $T^{\star}_{\mathbb{P}1\mathbb{L}\mathbb{P}}$. Thus, it is no more than $T^{\star}$. Starting from this time point on each PE, the task-PE mapping with the fractional x_{ijk} is determined by Part (b). Since at most m tasks get split in S, the task number in Part (b) is no more than m. Thus, similar to the case $n \leq m$, the maximum completion time of Part (b) is no more than $\mathrm{T}^{\star}$. Therefore, the overall makespan is no more than twice $\mathrm{T}^{\star}$. □

C. Scheduling on Heterogeneous CMP

Another kind of practical CMP architecture consists of a diversity of PEs, called heterogeneous (or asymmetric) CMP. Heterogeneous PEs imply different active v/f states with varying power/WCET characteristics. In this subsection, we propose a 2-approximation scheduling algorithm based on the algorithm for the generalized assignment problem (GAP) [18]. We construct a bipartite graph based on the schedule S generated from P1LP-OPT, achieve a minimum cost matching on the graph and then schedule the tasks according to the matching. The algorithm P_{asym} is described in Figure 3. Our technique differs from [18] in that our algorithm addresses the scheduling problem with one more dimension namely the active v/f state assignment.

In Step 2 of P_{asym} we construct a bipartite graph G with two disjoint node sets (U,V) and one edge set (E). One side of

P_{asym}:
Step 1: Achieve S from P1LP-OPT;
Step 2: Construct a bipartite graph $G = (U, V, E)$;
Step 3: Find a minimum cost matching A that exactly matches all the task nodes in G;
Step 4: For each edge in A, assign the task to the corresponding PE and the active voltage state via the associated x_{ijk}.

Fig. 3. A 2-approximation for EMMS problem with heterogeneous CMP

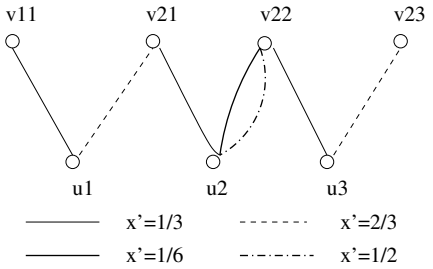

Fig. 4. The constructed bipartite graph $G(U, V, E)$

the graph, U, consists of all the task nodes $U = \{u_j | \tau_j \in \Gamma\}$. The other side of the graph, V, consists of all the PE nodes with $\sum_{j=1}^{n} \sum_{k=1}^{l_i} x_{ijk} > 0$. For each pe_i in V, there are $q_i = \lceil \sum_{j=1}^{n} \sum_{k=1}^{l_i} x_{ijk} \rceil$ nodes in V ($q_i \neq 0$).

Edges of the G are constructed based on the positive $\{x_{ijk}\}$ from the fractional schedule S. In this section, we only consider the items associated with $x_{ijk} > 0$ in S. Let v_{ih} denote the $h^{th}(h = 1, 2, ...q_i)$ node associated with pe_i in V. Let $e = (u_j, v_{ih})$ be an undirected edge connecting node u_j and v_{ih}. For each $pe_i \in M$, construct a list including all the t_{ijk} with positive $x_{ijk}, \forall \tau_j \in \Gamma, s_k \in \Psi_i$. Sort this list in the non-increasing order of t_{ijk}, and name the sorted list as $L_i(t) = \{t_{ijk}\}$. Construct an associated list $L_i(x) = \{x_{ijk}\}$ according to the order of $L_i(t)$. Recall that there are q_i nodes for the pe_i in V. If $q_i = 1$, construct an edge $e = (u_j, v_{i1})$ for every $x_{ijk} > 0$ and assign $x'(e) = x_{ijk}, t(e) = t_{ijk}, c(e) = c_{ijk}$. If $q_i > 1$, for v_{i1} ($h = 1$) find the smallest splitting index r in $L_i(x)$ such that $\sum_1^r x_{ijk} \geq 1$. Construct $r - 1$ edges $e = (u_j, v_{i1})$ for the first $r - 1$ x_{ijk} in $L_i(x)$. Assign $x'(e) = x_{ijk}$ as the case of $q_i = 1$. Add an edge for the r_{th} x_{ijk} as $e = (u_j, v_{i1})$ and assign $x'(e) = 1 - \sum_1^{r-1} x_{ijk}$. Delete the first $r - 1$ x_{ijk} from $L_i(x)$ and replace the r_{th} x_{ijk} as $\sum_1^r x_{ijk} - 1$. The assignment rules of $t(e)$ and $c(e)$ are always the same as those of $q_i = 1$ case. Similarly, for each $h = 2, 3, ..., q_i$, construct the edges and $x'(e)$ such that the following properties hold true:

i. $\forall v_{ih} \in V, \sum_{e \in E_{ih}} x'(e) = 1$, where E_{ih} denotes all the edges e incident to $v_{ih}, \forall h = 1, 2, ..., q_i - 1$.

ii. $\forall i \in M, \sum_{j=1}^{n} \sum_{k=1}^{l_i} x_{ijk} = \sum_{e \in E_i} x'(e)$, where E_i includes all the edges incident to any node of pe_i in V.

iii. $\forall i \in M, \min(t_{e \in E_{ih}}(e)) \geq \max(t_{e \in E_{ih+1}}(e))$.

Properties i. and ii. follow from the computation of $x'(e)$. Property iii. follows from edge construction based on $L_i(x)$ in non-increasing order of t_{ijk}.

We present an example to show the construction. Suppose that $m = 2, n = 3$. pe_1 only has one voltage, and pe_2 has two voltage states. After the LP relaxation of $\mathbb{P}1$ problem, the schedule S is a 3×3 matrix as follows. The values inside the square embraces [ ] are the related t_{ijk}.

$$\begin{pmatrix} x_{111} = \frac{1}{3}[6] & x_{211} = 0 & x_{212} = \frac{2}{3}[6] \\ x_{121} = 0 & x_{221} = \frac{1}{2}[5] & x_{222} = \frac{1}{2}[4] \\ x_{131} = 0 & x_{231} = 1[1] & x_{232} = 0 \end{pmatrix}$$

The constructed bipartite graph is shown in Figure 4. On pe_1, because $h_1 = \lceil x_{111} + x_{121} + x_{131} \rceil = 1$, $x'(u_1, v_{11}) = x_{111}$. On pe_2, because $\sum_{j=1}^{n} \sum_{k=1}^{l_i} x_{2jk} = 2\frac{2}{3}$, $h_2 = 3$. According to the non-increasing order of t_{ijk}, $L_2(t) = \{6, 5, 4, 1\}$. Therefore, $L_2(x) = \{x_{212}, x_{221}, x_{222}, x_{231}\}$. The first splitting item is x_{221} in $L_2(x)$ as $x_{212} + x_{221} > 1$. We add edge $(u1, v21)$ with $x'(e) = x_{212}$. Then, for x_{221}, we add an edge $(u2, v21)$ and assign $x'(e) = 1 - x_{212} = \frac{1}{3}$. We delete x_{212} from $L_2(x)$ and replace x_{221} as $x_{212} + x_{221} - 1 = \frac{1}{6}$. Thus, $L_2(x) = \{\frac{1}{6}, x_{222}, x_{231}\}$. For $v22$, because $\frac{1}{6} + x_{222} + x_{231} > 1$, we add edges $(u2, v22)$ and $(u2, v22)$ with $x'(e) = \frac{1}{6}$ and $x'(e) = x_{222} = \frac{1}{2}$ respectively. We construct one portion of x_{231} as edge $(u3, v22)$ with $x'(e) = \frac{1}{3}$ and another portion as edge $(u3, v23)$ with $x'(e) = \frac{2}{3}$. The resulting bipartite graph embeds the mentioned properties.

Observe that a minimum cost matching on the graph G is actually a feasible solution of the scheduling if the total energy cost is no more than C. In Figure 3, Step 3 of the P_{asym} algorithm computes the minimum cost matching in the bipartite graph G. Based on the properties of G, a maximum flow matching in G is a schedule for the EMMS problem. For example in Figure 4, the corresponding $(x_{111}, x_{221}, x_{231})$ of the matching $\{(u1, v11), (u2, v21), (u3, v22)\}$ is a schedule for the EMMS problem. To satisfy the energy budget constraint, we should find the minimum cost matching (MCM) on G, where the cost on any edge stands for the energy cost $c(e)$. R. Ahuja et al. [2] have shown that any basic feasible solution of the LP relaxation of the MCM problem is integral.

Lemma 2. *The minimum cost matching exists in $G(U, V, E)$ and the energy consumption of the matching is at most C.*
Proof. Because the fractional vector $x'(e)$ is a feasible solution of the LP relaxation of the MCM problem, and the minimum cost is no more than C, it is an upper bound of the optimal LP relaxation problem. According to [2], the basic feasible solution of the LP relaxation would be integral. Thus, the minimum cost matching exists and the total energy consumption is at most C. □

Theorem 3. *The makespan of the schedule generated from the P_{asym} algorithm is at most as twice as the optimal.*
Proof. The proof is similar to that in [18]. In the minimum cost matching A, there is at most one task scheduled on each node of the pe_i in V. Therefore, the makespan of the matching $T(A) \leq \sum_{h=1}^{q_i} \max(t_{e \in E_{ih}}(e))$. For the first node $v_{i1}(h = 1)$, $\max(t_{e \in E_{i1}}(e)) \leq T^\star$, because $t(e) = t_{ijk}$ in G and Constraint (2) is satisfied. For the remaining nodes of pe_i,

$$\sum_{h=2}^{q_i} \max(t_{e \in E_{ih}}(e)) \leq \sum_{h=1}^{q_i - 1} \min(t_{e \in E_{ih}}(e))$$
$$\leq \sum_{h=1}^{q_i - 1} \sum_{e \in E_{ih}} t(e)x'(e) \leq \sum_{h=1}^{q_i} \sum_{e \in E_{ih}} t(e)x'(e)$$
$$= \sum_{j=1}^{n} \sum_{k=1}^{l_i} t_{ijk} x_{ijk} \leq T^\star$$

The first inequality follows from Property iii. of the graph G. The second inequality follows from the definition of

$\min\left(t_{e\in E_{ih}}(e)\right)$ and Property i. of the graph G. The fourth equality follows from the construction of $x'(e)$ and $t(e)$. The last inequality follows from the Constraint (1b) of the $\mathbb{P}2\mathbb{L}\mathbb{P}$. Thus, P_{asym} is a 2-approximation algorithm. □

D. Complexity Analysis

Since P1LP-OPT performs Step 1 in both P_{sym} and P_{asym} algorithms, the computational complexity of P1LP-OPT influences the complexity of the proposed techniques. Simplex method is a well-known polynomial time algorithm to solve linear programming problems. Let C_o denote the computational complexity of the simplex method, which is polynomial. The computation complexity of the P1LP-OPT is $O(log(\frac{T_{UB}}{T_{LB}})C_o)$, because of the binary search.

Let $l = \max_{\forall i\in M}\{l_i\}$. In the P_{sym} algorithm, Step 1 dominates the overall complexity. Step 2 and Step 3 only take at most $O(nml)$. Thus, it is a polynomial time algorithm. In the P_{asym} algorithm, Step 1 is polynomial as discussed above. In Step 2, because the schedule S consists of at most $n+m$ positive x_{ijk}, the the sorting algorithm for the list $L_i(t)$ takes at most $O((n+m)\log(n+m))$ time by merge sorting for each PE. Step 3 is of polynomial complexity as it utilizes the simplex algorithm on the MMS problem. Step 4 is at most $O(nml)$. Thus, the computational complexity of P_{asym} algorithm is polynomial.

V. Results

We evaluated the proposed techniques with extensive experiments that are presented in this section. In the case of homogeneous and heterogeneous CMPs, we analyzed the achieved approximation ratio with the effects of two factors : the CMP architecture and the task patterns. We compared the makespan generated from P_{sym}, P_{asym}, the ILP solver from [1] and the tight lower bound of $\mathbb{P}1$ (P1LP-OPT). In some cases, the ILP solver took an unbounded large amount of time to achieve an optimal. We set a timeout of 10000 seconds, after which the ILP solver returned the best suboptimal solution. In all the plots, the makespan values were normalized with respect to the P1LP-OPT (the tight lower bound of the EMMS problem), which can directly reflect the actual approximation ratio[1]. The runtimes of the proposed techniques was also studied in comparison to the ILP solver with 8 hours timeout configuration.

A. Experimental Setup

We obtained the PE models from two commercial DVFS-equipped processors: IBM PowerPC [10] and Intel PXA270 [12]. We chose 6 v/f states for PowerPC ranging from 1V/1.0GHz to 1.25V/2.0GHz and 7 v/f states for PXA270 ranging from 0.85V/13MHz to 1.55V/624MHz. For homogeneous CMPs, the PowerPC was set as the PE unit to compose the multiple PE system. For heterogeneous CMPs, four combinations of the PowerPC and the PXA-270 were chosen as the target CMPs. We designed 4 task sets with different workload distributions: *equal, uniform, Gaussian* and *Poisson*. Each task set included 30 task nodes. The cycle number of the tasks were in the range of $[10^6, 10^{10}]$. For the energy budget, a metric named energy budget ratio $r(r \in [0,1])$ from [9] was introduced. With various r values, we set $\mathrm{C} = \sum_j (r\cdot(\max_{i,k}\{c_{ijk}\} - \min_{i,k}\{c_{ijk}\}) + \min_{i,k}\{c_{ijk}\})$, where $pe_i \in \Phi, \tau_j \in \Gamma, s_k \in \Psi_i$. The optimization techniques were coded in C++ and the experimentations were performed on a Pentium 4/2.4GHz/1GB WindowsXP PC.

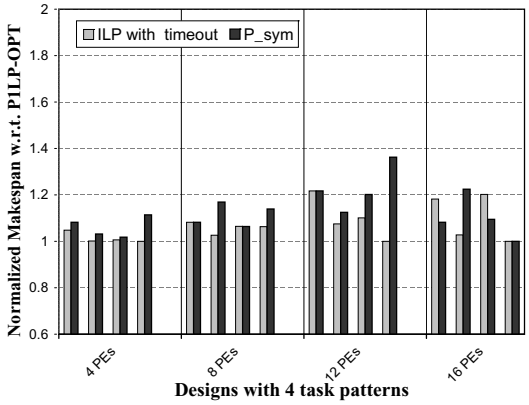

Fig. 5. Evaluation on Homogeneous CMP architecture

B. Effect of CMP architecture and task patterns

We evaluated the proposed techniques by experimenting with the 4 task patterns. For the homogeneous CMP case, the number of PEs were varied from 4 to 16. For the heterogeneous CMP case, we designed four kinds of CMP with combinations of multiple PowerPC and PXA270. For the both cases, we compared the makespan generated from P_{sym}, P_{asym} and the ILP solver which are plotted in Figures 5 and 6. All the makespan values are normalized to the tight lower bound of $\mathbb{P}1$ generated from P1LP-OPT. Therefore, the actual approximation ratio is no more than the normalized makespan. Each CMP was plotted as a separate category. In each category, the results from 4 task sets were depicted from left to right in the order of equal, uniform, Gaussian and Poisson. The energy budget ratio was set as 0.5.

Homogeneous CMP The four target CMPs were designed as CMPs with 4, 8, 12, and 16 PowerPC PEs. As observed in Figure 5, the normalized makespan generated from P_{sym} is no more than 1.36 with all the task sets. With the 16 PEs, the results are better than ILP solver for the task sets with equal/Gaussian workload. In all cases, the average approximation ratio to the P1LP-OPT is 1.13, while the average ratio to the ILP is 1.06. With each task pattern, the average approximation ratio to the P1LP-OPT is below 1.15.

Heterogeneous CMP We designed 4 types of heterogenous CMPs with PowerPC and PXA270. We denote the PowerPC as *H* and the PXA270 as *L*. The 4 heterogeneous CMPs are plotted in the following order *1H3L, 1H8L, 1H16L, 2H16L*. As shown in Figure 6, the normalized makespan generated from P_{asym} (the upper bound of the actual approximation ratio) is within the theoretical bound of 2. In general, the normalized makespan for heterogeneous CMP is larger than that for the

[1]The actual approximation ratio is no more than the normalized makespan w.r.t the P1LP-OPT. Even if there is an integrality gap between the LP relaxation and ILP problem, the normalized makespan is still meaningful. This is because the P1LP-OPT is a tight lower bound of the ILP problem.

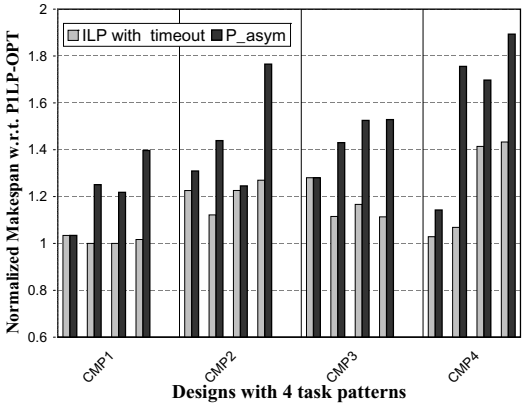

Fig. 6. Evaluation on Heterogeneous CMP architecture

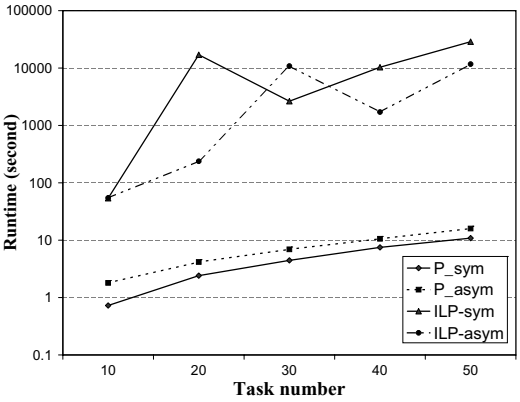

Fig. 7. Runtime versus task number

homogeneous CMP with all the task patterns. For the normalized makespan generated from the P_{asym}, the maximum ratio with comparison to the ILP is 1.64. In all the cases, the average normalized makespan of the P_{asym} is 1.43, while that of the ILP is 1.24.

Summary The actual approximation ratios of the schedules generated by P_{sym} and P_{asym} are within the theoretical bound. The task patterns have less effect on solution quality for the homogeneous CMP than that for the heterogeneous CMP.

C. Runtime

To evaluate the complexity of our techniques, synthetic task sets with up to 50 nodes and uniformly distributed runtimes were generated. The CMP with 4 PowerPC PEs was targeted for the homogeneous case and the *1H3L* CMP was targeted for the heterogeneous case. The energy budget ratio was set as 0.5. We compared the average runtime of P_{sym}, P_{asym} with the ILP solver (8 hours timeout setting) in Figure 7. The number of tasks in the task sets was varied from 10 to 50 nodes in steps of 10. Note that the y axis is in logarithmic scale in Figure 7. With up to 50 nodes, the P_{sym} and P_{asym} algorithms were completed within half a minute. The figure shows that the runtime of our techniques is linearly increasing with the increase of task numbers. As predicted, the P_{sym} is slightly faster than the P_{asym} algorithm because of the simplicity of the former. In comparison, the runtime of the ILP solver is exponentially large in some cases. Even with 10 nodes, the average runtime of the ILP solver is around 10 times of the P_{sym} and P_{asym}. With 50 nodes, the average runtime is beyond 8 hours and is actually more than 1000 times of our techniques. Therefore, the results demonstrate that the proposed techniques are efficient, and applicable in practice.

VI. Conclusion

In this work, we addressed the energy-efficient scheduling problem on CMP architectures with core-level DVFS. We proved that the EMMS problem is strongly NP-hard. We then proposed 2-approximation polynomial time techniques for both homogeneous and heterogeneous CMP. Our extensive experimentation with multiple workloads and CMP architectures demonstrate that our techniques can efficiently generate solutions whose makespan is much lower than the factor of 2 in comparison to the optimal that is guaranteed by the approximation bound.

References

[1] Lp/ilp solver. *http://lpsolve.sourceforge.net/5.5/.*

[2] R. Ahuja, T. Magnanti, and J. Orlin. Network flows: Theory, algorithms and applications. *Prentice Hall*, 1993.

[3] A. Andrei, P. Eles, and Z. Peng. Energy optimization of multiprocessor systems on chip by voltage selection. *IEEE Trans. on VLSI Systems*, 15(3):262–275, 2007.

[4] H. Aydin and Q. Yang. Energy-aware partitioning for multiprocessor real-time systems. In *Proc. of IPDPS*, 2003.

[5] D. Bunde. Power-aware scheduling for makespan and flow. In *Proc. of ACM symposium on parallelism in algorithms and architectures*, 2006.

[6] J. Chen, C. Yang, T. Kuo, and C. Shih. Energy-efficient real-time task scheduling in multiprocessor dvs systems. In *Proc. of ASPDAC*, 2007.

[7] A. Eichenberger, J. O'Brien, and et al. Using advanced compiler technology to exploit the performance of the cell broadband engine^{TM} architecture. *IBM Systems Journal*, 45:59–84, 2006.

[8] D. Hochbaum. Approximation algorithms for np-hard problems. *PWS Publishing Company*, 1997.

[9] H. Hsu, J. Chen, and T. Kuo. Multiprocessor synthesis for periodic hard real-time tasks under a given energy constraint. In *Proc. of DATE*, 2006.

[10] IBM. Ibm powerpc 970fx risc microprocessor datasheet. 2007.

[11] Intel. Intel ixp2855 network processor - product brief.

[12] Intel. Intel pxa270 processor: electrical,mechanical, and thermal specifi. 2005.

[13] C. Isci, A. Buyuktosunoglu, C-Y. Cher, P. Bose and M. Martonosi. An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget. *IEEE/ACM International Symposium on Microarchitecture*, 2006.

[14] N.K. Jha. Low power system scheduling and synthesis. In *Proc. of ICCAD*, 2001.

[15] J. Li and J. Martinez. Dynamic power-performance adaptation of parallel computation on chip multiprocessors. In *Proc. of HPCA*, 2006.

[16] C. Papadimitriou and K. Steiglitz. Combinatorial optimization: algorithms and complexity. *Dover Publications*, 1998.

[17] K. Pruhs, R. van Stee, and P. Uthaisombut. Speed scaling of tasks with precedence constraints. In *Proc. of the 3rd workshop on approximation and online algorithms*, volume 3879 of LNCS, 2005.

[18] D. Shmoys and E. Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62:461–471, 1993.

[19] G. Varatkar and R. Marculescu. Communication-aware task scheduling and voltage selection for total systems energy minimization. In *Proc. of ICCAD*, 2003.